

# Sentiment Analysis of IMDB movie reviews.



---

The dataset contains two columns:




**text** - Movie reviews

**label** - Classification of the reviews as positive (1) and negative(0)

## Dataset first 5 rows

	text	label	
0	Very silly movie, filled with stupid one liner...	0	
1	As predictable as a Hallmark card, but not wit...	1	
2	Only a 9/10 from me, a perfect ten would have ...	1	
3	After Watergate, Vietnam and the dark days of ...	0	
4	As long as you keep in mind that the productio...	0	

## Dataset last 5 rows

	text	label	
37925	I have seen this film many times and I like al...	1	
37926	When I saw it for the first time I was really ...	1	
37927	the reason why i gave this movie a 4 was for a...	0	
37928	I'd have to admit that the draw of this movie ...	0	
37929	drab morality tale about a high school kid who...	0	

## Dataset Info

```
➞ <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 37930 entries, 0 to 37929  
Data columns (total 2 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   text    37930 non-null  object  
1   label    37930 non-null  int64  
dtypes: int64(1), object(1)  
memory usage: 592.8+ KB
```

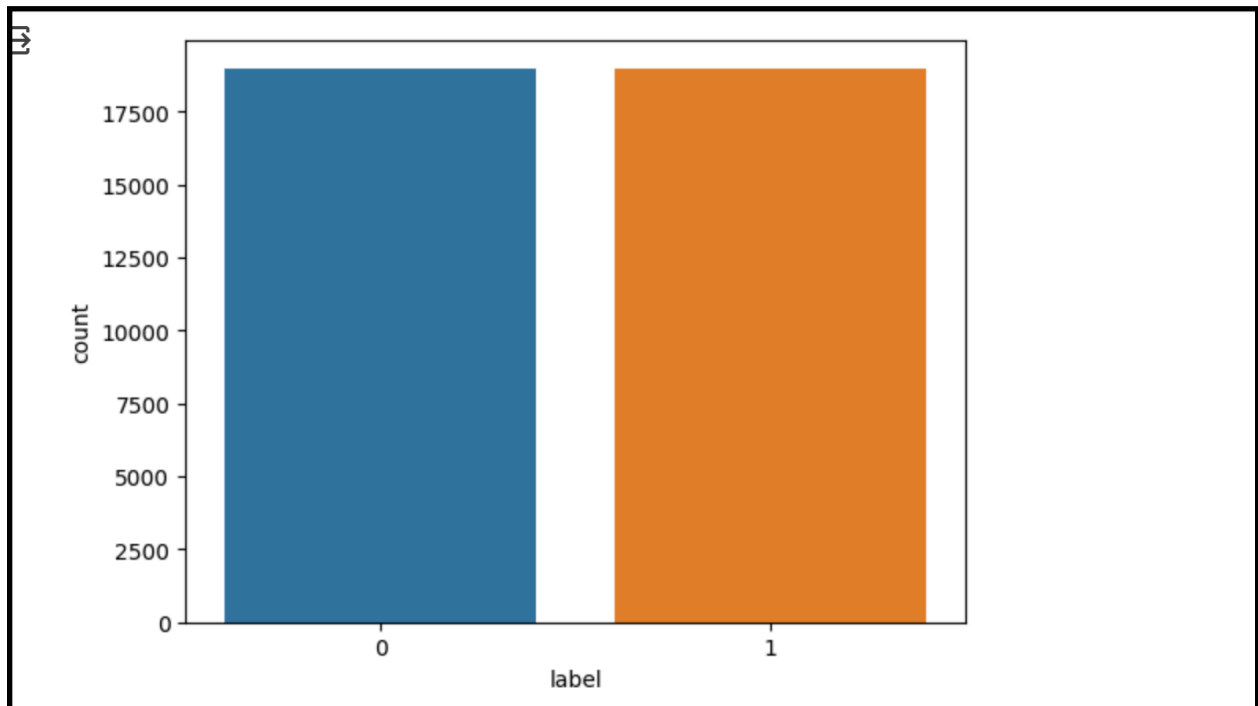
## No Missing Values

```
➞ text    0  
   label    0  
   dtype: int64
```

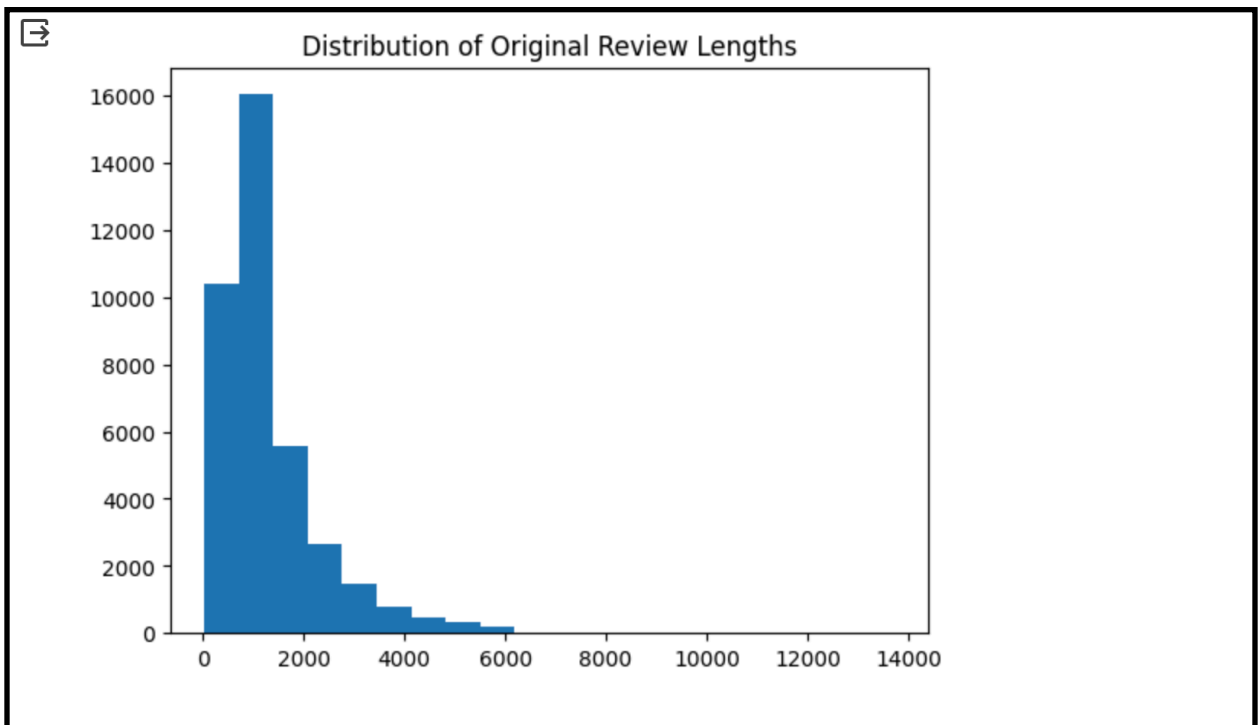
## Dataset size and Label distribution

```
➞ Dataset size: 37930  
   Label distribution:  
    0    18978  
    1    18952  
   Name: label, dtype: int64
```

### Sentiment distribution (0 - negative, 1 - positive)




### Plot of the length of the review





### **Data Cleaning to remove the HTML characters, special characters like punctuation and numbers. Also, converting the text to lowercase**

## Cleaned data



	cleaned_text	label
0	very silly movie filled with stupid one liners...	0
1	as predictable as a hallmark card but not with...	1
2	only a from me a perfect ten would have been ...	1
3	after watergate vietnam and the dark days of t...	0
4	as long as you keep in mind that the productio...	0



## Word Cloud



## Bigram 10 most common words in the cleaned review text data

```
[('of', 'the'), 58606], (('in', 'the'), 37749), (('this', 'movie'), 22455), (('and', 'the'), 19948), (('is', 'a'), 19737), (('to', 'the'), 17919), (('to', 'be'), 17680), (('the', 'film'), 17661), (('the', 'movie'), 16622), (('this', 'film'), 15453)]
```



## Text Preprocessing

**Tokenization:** Splitting text into individual words.

**Removing Stop Words:** Eliminating common words that may not add much meaning to the analysis.

**Lemmatization:** Reducing words to their base or dictionary form.

## Preprocessed text

	preprocessed_text	label	
0	silly movie filled stupid one liner jewish ref...	0	
1	predictable hallmark card without merit rookie...	1	
2	perfect ten would plot movie nevertheless moon...	1	
3	watergate vietnam dark day nixon jimmy carter ...	0	
4	long keep mind production movie copyright ploy...	0	

## Bigram 10 most common words in the preprocessed review text data

```
[('look', 'like'), 2237], (('ever', 'seen'), 1944), (('special', 'effect'), 1664), (('ive', 'seen'), 1646), (('dont', 'know'), 1546), (('main', 'character'), 1426), (('even', 'though'), 1424), (('one', 'best'), 1401), (('movie', 'like'), 1311), (('year', 'old'), 1306)]
```

## Feature Extraction

TF-IDF for feature extraction. It involves converting text data to numerical form.

Dimension of the resulting feature matrix  
(37930, 170028)

## Model Training

Split the data in training and test set to evaluate the performance of the model.  
We used the Logistic Regression model.

## Model Evaluation

Classification Report:				
	precision	recall	f1-score	support
0	0.90	0.88	0.89	3784
1	0.88	0.90	0.89	3802
accuracy			0.89	7586
macro avg	0.89	0.89	0.89	7586
weighted avg	0.89	0.89	0.89	7586
Confusion Matrix:				
[[3317 467]				
[ 377 3425]]				
Accuracy Score:				
0.8887424202478249				

**Precision :** For the label 0 (negative review) the model has **correctly** predicted 90% of the time and 88% of the time for label 1(positive review)

**Recall :** The model correctly identified 88% of all actual label 0 (negative reviews) instances and 90% of label 1(positive reviews) instances.

**F1 score :** The F1 score indicates that there is strong overall performance in both precision and recall.

**Support :** Both the labels have almost equal representation, with 3784 instances of label 0 and 3802 instances of label 1.

**Confusion Matrix:** The confusion matrix shows the number of correct and incorrect predictions. Out of 7586 instances, 3317 were correctly identified as label 0, and 3425 as label 1. There were 467 false positives (label 0 incorrectly identified as label 1) and 377 false negatives (label 1 incorrectly identified as label 0).

**Accuracy Score:** The overall accuracy of the model is 0.8887, or about 88.87%. This means that the model correctly predicts the labels for approximately 88.87% of the cases in the dataset.

---

## Analyzing the sentiments of the movie reviews generated by Chat GPT

Got a set of movie reviews data generated by chatGPT

Cleaned and preprocessed the data using the same cleaning steps as above.  
Then used the same TF-IDF vectorizer for feature extraction.

Used the trained model to predict the sentiment of the reviews.

### Results:

The results are accurate.

	movie-reviews	predicted_sentiment
0	Just watched 'Eternal Echoes' - a stunning ble...	1
1	Caught the latest superhero flick, 'Guardians ...	1
2	If you're looking for a heartwarming story, 'T...	1
3	Saw 'Mystery at the Manor' last night. The twi...	1
4	I'm still in awe of 'Stars Beyond'. The cinema...	1
5	Laughed non-stop during 'Comedy Central'. It's...	1
6	The action in 'Fist of Fury: Reloaded' is top-...	1
7	As a fan of the book, 'Worlds Apart' didn't qu...	0
8	Just left the theater from 'Dreams of Tomorrow...	1
9	Director Jane Smith's latest, 'Echoes of the P...	1

---

## Comparing the realism of the generated reviews and the real reviews using perplexity (code in colab)

```
⇒ Perplexity of Real Tweets: [29.03848648071289]  
   Perplexity of Generated Tweets: [29.03848648071289]
```