SENTIMENT ANALYSIS

Objective

To develop a model that can accurately classify text data as expressing positive or negative sentiment using the Naive Bayes algorithm, leveraging its simplicity and efficiency in handling text classification tasks.

About the dataset

The dataset is taken from Kaggle which includes IBM movie reviews.

Programming language

Used R Programming to build the model.

Analysis

Getting dataset

```r
data=read.csv("C:\\Users\\ASUS\\Downloads\\Datasets\\IMDB Dataset.csv")
```

Tokenization and Text Cleaning

```r
library(tidytext)
library(dplyr)
library(textstem)
library(stringr)
```

Removing stop-words

```r
data1=data %>%
  unnest_tokens(word,review) %>%
  anti_join(stop_words)
```

Removing punctuation and numbers

```r
data2=data1 %>%
  mutate(word = str_remove_all(word,"[^a-zA-Z]")) %>% #Removing non-alphabet characters
  filter(word!="" & word!="br" & word!="movie" & word!="film") #Removing empty and unnecessary words
```

Performing stemming

```r
data3=data2 %>%
  mutate(word=lemmatize_words(word))
```

Getting idea about the frequent words

```r
word_counts=data3 %>%
  count(word,sort=TRUE)
#word_counts
```

Splitting subsets for training part to train model and testing part for prediction

```r
#Training 70% 0f data and testing rest 30%
set.seed(1)
N=length(data3$sentiment)
n=0.7*N
train=sample(N,n)
train_data=data3[train, ]
test_data=data3[-train, ]
```

Fitting the model and making prediction

```r
require(e1071) #Loading package for naiveBayes function
```

```
model=naiveBayes(sentiment~word,data=train_data)
prediction=predict(model,newdata=test_data)
```

Getting the result

```
Confusion_Matrix=table(prediction,test_data$sentiment)
Accuracy=mean(prediction==test_data$sentiment)
#Output
Confusion_Matrix


 prediction\Actual negative positive
   negative         317382   225097
   positive         290428   417476

Accuracy
## 0.5877063
```

Results

1. **Accuracy**: Achieved **58%** accuracy in classifying text data into positive or negative sentiments.

   o Positive sentiments correctly classified=65%
   [417476/(417476+225097)≈0.65]

   o Positive sentiments correctly classified=52%
   [317382/(317382+290428)≈0.52]

2. **Performance**: Naive Bayes performed moderately well but was sensitive to text preprocessing and feature engineering.

3. **Insights**: Highlighted the algorithm's strength in handling high-dimensional text data but showed limitations with complex sentiment nuances due to its assumption of feature independence.

Conclusion

The Naive Bayes model provided a baseline performance for sentiment analysis and demonstrated the importance of advanced preprocessing for improved accuracy.