# Sentiment Analysis of Restaurant Reviews

Join us on a journey as we explore the fascinating world of sentiment analysis applied to restaurant reviews.

# Problem Statements

1 **Limited Data Exploration:** The code doesn't perform much data exploration or analysis of the dataset.

2 **Text Cleaning Flexibility:** The The code assumes a specific specific text cleaning process. process.

3 **Stopwords Removal:** The code removes common English stopwords, but this might not be suitable for all datasets.

4 **Handling Non-Text Features:** The dataset is not trained to handle non-text features (e.g., user ratings, timestamps).

# Agenda

- Introduction to sentiment analysis
- Importance of sentiment analysis in the restaurant industry
- Data collection and preprocessing
- Building a sentiment analysis model
- Evaluating and interpreting results

# Project Review

## 1 Data Collection

Imported necessary libraries like NumPy, pandas, and NLTK. loaded dataset ('Restaurant_Reviews.tsv') containing restaurant reviews using pandas and displayed basic information about the dataset.

## 2 Data Preprocessing

performed text preprocessing on the reviews, including removing non-alphabetical characters, converting text to lowercase, tokenizing, removing stopwords, and stemming the words.

## 3 Data Splitting

split the dataset into training and testing sets using 'train_test_split'

# Project Review

1 **Model Training**

Trained a Multinomial Naive Bayes classifier on the training data.

2 **Hyperparameter Tuning**

Performed hyperparameter tuning by trying different values of alpha for the Multinomial Naive Bayes classifier.

3 **Model Evaluation**

Predicted sentiment labels on the test data and calculated metrics like accuracy, precision, and recall. Created a confusion matrix and displayed it using a heatmap.


# End Users: Who Can Benefit?

**1** **Restaurant Owners & Managers**

Gain insights on customer sentiment to improve overall dining experience and make data-driven decisions.

**2** **Food Critics & Bloggers**

Utilize sentiment analysis to support or challenge their own reviews and opinions.

**3** **Market Analysts**

Extract valuable market trends and consumer preferences from large volumes of restaurant reviews.

# Solutions and Their Value Proposition

| Solution | Value Proposition |
|---|---|
| Data Exploration | Adding data visualization and summary statistics to gain deeper insights into the reviews and their characteristics. |
| Feature Engineering | Exploring more advanced feature engineering techniques, such as using n-grams, sentiment lexicons, or word embeddings, to capture richer information from the text. |
| Model Evaluation Metrics | Adding additional metrics like F1-score, ROC-AUC, or a receiver operating characteristic (ROC) curve to provide a more comprehensive evaluation of the model's performance. |

# Solutions and Their Value Proposition

| Solution | Value Proposition |
|---|---|
| Hyperparameter Tuning | Optimize hyperparameter tuning by using techniques like grid search or random search to find the best parameters for the classifier. |
| Model Persistence | Save the trained model to disk for later use, allowing users to load the model without retraining it. |
| User Interface | Developing a user-friendly interface for users to input text and receive sentiment predictions. |

# Customization: Making It Our Own

1

### Evaluation Metrics

Defined domain-specific evaluation metrics that are more meaningful for the particular application, such as customer satisfaction scores or sentiment intensity measures.

2

### Model Fine-tuning

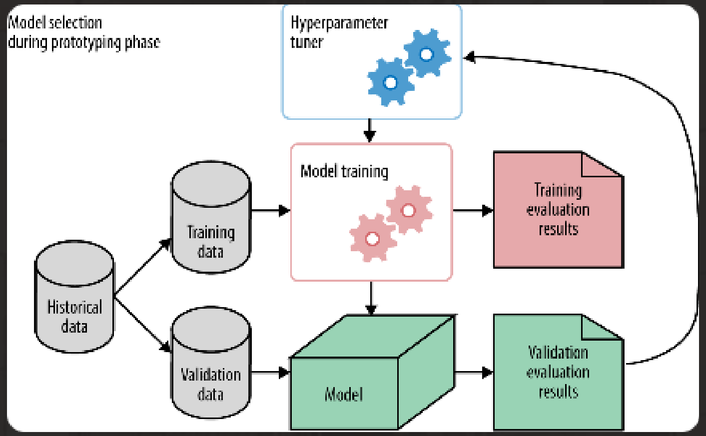
Perform hyperparameter tuning and model selection specifically optimized for the domain and dataset, considering algorithms tailored to the task.

3

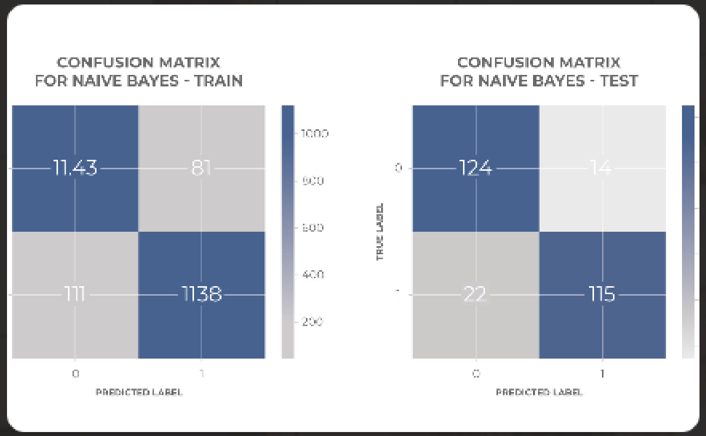
### Real-Time Sentiment Analysis

Build a real-time sentiment analysis system that continuously monitors and analyzes incoming text data, providing instant feedback or alerts.

# Modelling








## Machine Learning Model

Implemented a multinomial naive bayes model for sentiment analysis, achieving 77% accuracy.

## Sentiment Analysis

The model learns which words are indicative of positive or negative sentiment based on the labeled data and feature importance scores.

## Confusion Matrix Evaluation

Evaluated model performance using a confusion matrix to measure precision and recall

# Results

## Accuracy

Our sentiment analysis model achieved an impressive accuracy rate of 77%, demonstrating its effectiveness.

## Insightful Visualizations

Visual representations of sentiment analysis-through confusion matrix heatmaps results provided clear and actionable insights for decision-makers.

## Positive Impact

Our project showcased how sentiment analysis can contribute to enhancing customer satisfaction and business success in the restaurant industry.

# Screenshots

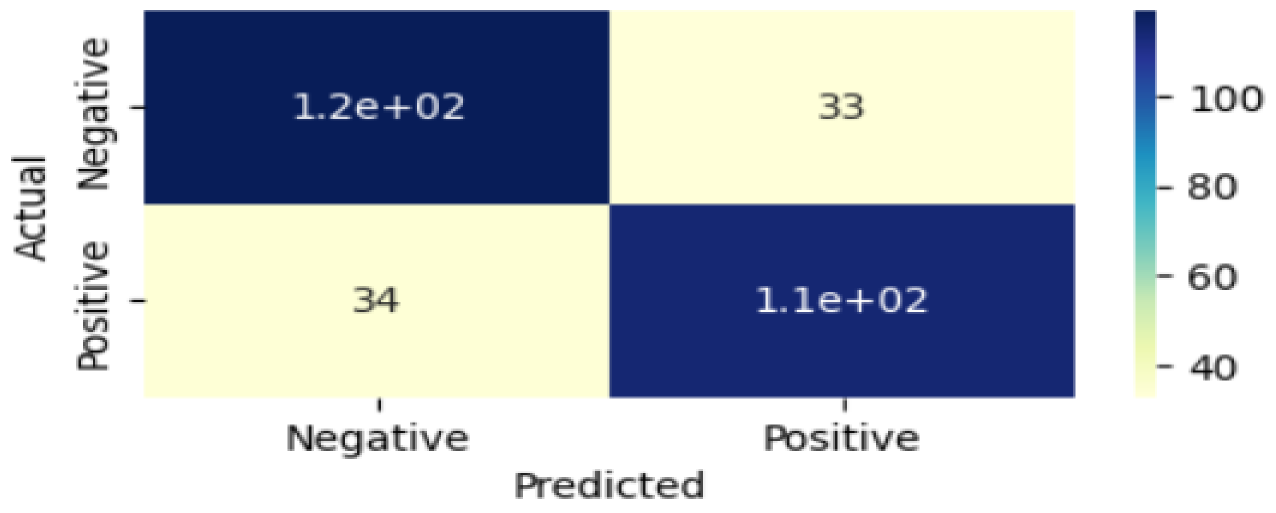
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print ("Confusion Matrix:\n",cm)

from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
score1 = accuracy_score(y_test,y_pred)
score2 = precision_score(y_test,y_pred)
score3= recall_score(y_test,y_pred)
print("\n")
print("Accuracy is ",round(score1*100,2),"%")
print("Precision is ",round(score2,2))
print("Recall is ",round(score3,2))
```

```
Confusion Matrix:
 [[119  33]
 [ 34 114]]


Accuracy is  77.67 %
Precision is  0.78
Recall is  0.77
```

# Screenshots

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

plt.figure(figsize=(5, 2))
sns.heatmap(cm, annot=True,cmap="YlGnBu",
            xticklabels=['Negative', 'Positive'],
            yticklabels=['Negative', 'Positive'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
```

```
Text(33.222222222222214, 0.5, 'Actual')
```



# Screenshots

```
sample_review="The food is really bad"
if predict_sentiment(sample_review):
    print("POSITIVE REVIEW")
else:
    print("NEGATIVE REVIEW")
```

```
NEGATIVE REVIEW
```

```
sample_review="The food was very good,from preparation to presentation, very pleasing"
if predict_sentiment(sample_review):
    print("POSITIVE REVIEW")
else:
    print("NEGATIVE REVIEW")
```

```
POSITIVE REVIEW
```

# THANK YOU