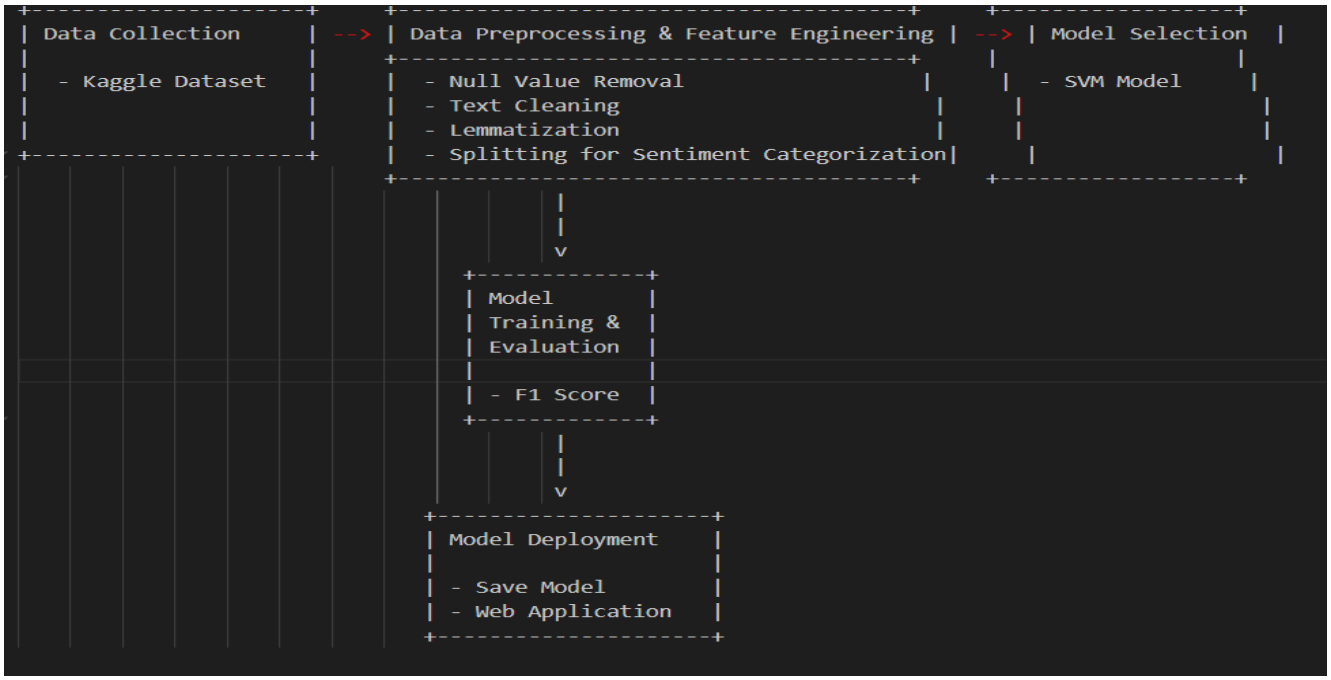
3.2 Problem Statement:

Build a sentiment analysis model to analyze customer reviews and feedback for a hospitality company. The model should be able to classify reviews as positive, negative, or neutral, and identify key topics or areas of concern.

Introduction

Customer feedback plays a pivotal role in shaping the hospitality industry, where service quality directly impacts customer satisfaction and loyalty. Manual analysis of feedback can be time-consuming and prone to errors, especially with large volumes of data. To address this challenge, we propose a sentiment analysis model that automates the classification of customer reviews as positive, negative, or neutral, providing valuable insights for improving services.

Work Flow process



Dataset

The dataset I used for this analysis is sourced from Kaggle and contains customer reviews of a hospitality company. It includes features such as review text and star ratings, which serve as the basis for sentiment classification.

Libraries Used

- Pandas
- matplotlib/Seaborn
- Wordcloud
- Nltk
- Sklearn
- Joblib
- Flask

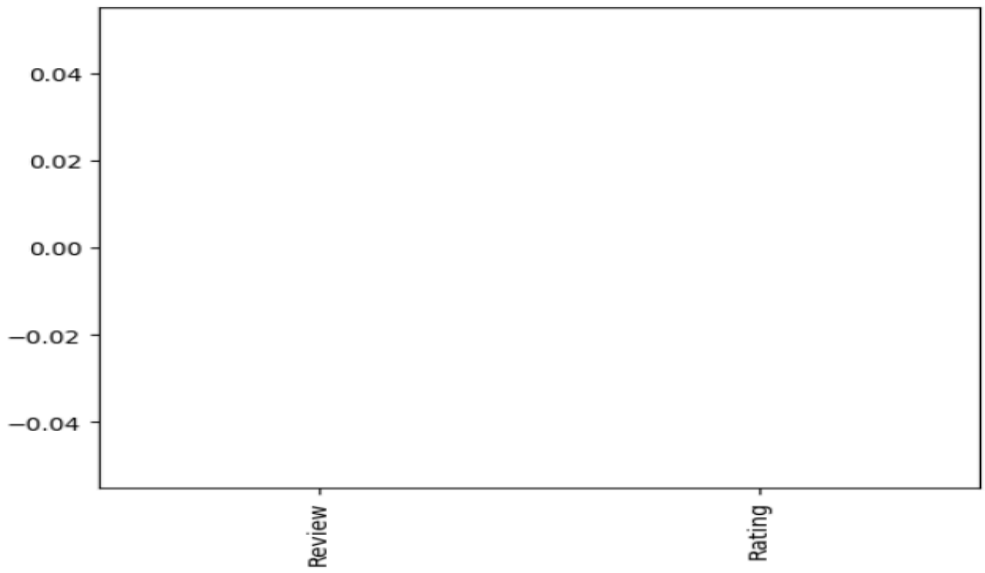
Preprocessing

Examination: Understanding the dataset's structure and features is pivotal for effective preprocessing and analysis.
Cleaning: Removing null values ensures the dataset's integrity and prevents bias during analysis.

```
def analysing_data_attributes(x):
    return x.isna().sum().plot(kind = 'bar')
```

```
analysing_data_attributes(df)
```

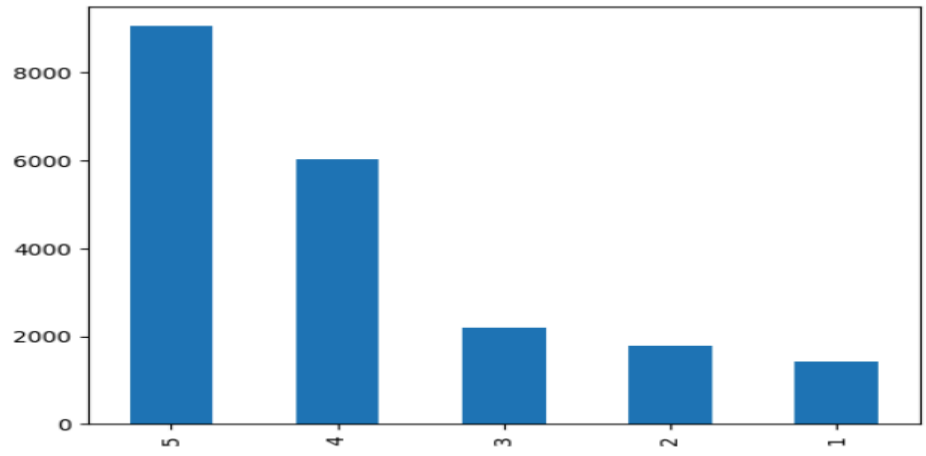
```
<Axes: >
```



The dataset is free from null values.

```
df['Rating'].value_counts().plot(kind = 'bar')
```

```
<Axes: >
```



logic building

- if the value below 3 , cosider the review to be negative. i.e 0
- if the value above 3 or equal to 3, consider the review to be poitive. 1

From the above figure, I can see that the majority of the customers provided positive feedback regarding the hospitality company.

Splitting: Categorizing reviews based on star ratings (e.g., considering reviews below 3 stars as negative) simplifies the sentiment analysis task.

```
df['Feedback'] = df['Rating'].apply(lambda x: 1 if x >=3 else 0)
```

```
df.head()
```

	Review	Rating	Feedback
0	nice hotel expensive parking got good deal sta...	4	1
1	ok nothing special charge diamond member hilto...	2	0
2	nice rooms not 4* experience hotel monaco seat...	3	1
3	unique, great stay, wonderful time hotel monac...	5	1
4	great stay great stay, went seahawk game aweso...	5	1

Analysing the Occurance of Text Data Using WordCloud

Word Cloud: Visualizing word frequencies using a word cloud helps identify common themes and sentiments expressed in customer feedback.

```
from wordcloud import WordCloud
df = df[df['Review'].notna()]
text = ' '.join(df['Review'])
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Preprocessing text

The model can’t able to understand the textual data. So , converting the text data into embedding /vectors. Such a model can understand and provide forecasting.

Normalization: Converting text to lowercase and removing stop words standardizes the text data for further analysis.

Lemmatization: Transforming words to their base form using lemmatization improves the accuracy of sentiment analysis by capturing the true meaning of words.

```
def preprocess(raw_text):
    sentence = re.sub("[^a-zA-Z]", " ", raw_text)

    sentence = sentence.lower()

    tokens = sentence.split()

    clean_tokens = [t for t in tokens if t not in english_stopwords]

    clean_tokens = [lemmatizer.lemmatize(word) for word in clean_tokens]

    return " ".join(clean_tokens)

X_train = x_train.apply(preprocess)
```

Converting Sentences to Vectors

Bag of Words: Converting text data into numerical vectors using the bag of words technique enables machine learning models to process and analyze the data effectively.

Model Selection

Pipeline: Creating a pipeline that combines preprocessing and model training streamlines the analysis process.

Models: Experimenting with various models such as Support Vector Machine, Logistic Regression, Naive Bayes, and Decision Tree helps identify the best-performing model for sentiment analysis.

The Support Vector Machine (SVM) model you've finalized with an F1 accuracy score of 0.9611 is performing exceptionally well. Let's elaborate on the choice of hyperparameters and kernel:

Choice of Hyperparameter :

1. The regularization parameter C controls the trade-off between a smooth decision boundary and classifying the training points correctly.
2. A smaller value of C (e.g., 0.1) allows for a softer margin, which can help generalize better to unseen data and reduce overfitting.
3. A larger value of C (e.g., 10) leads to a harder margin, which may fit the training data more closely but

Model Deployment

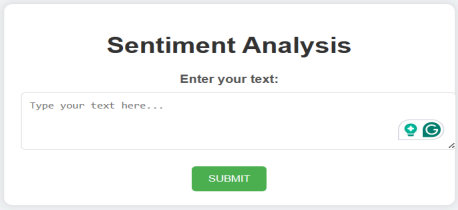
Saving the trained model using joblib ensures its reusability and accessibility for future analysis.

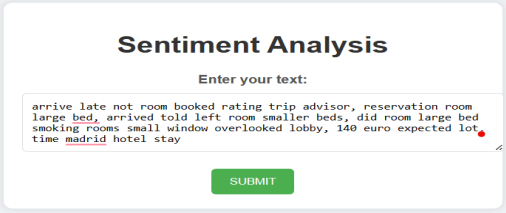
Web Application: Deploying the model as a web application using Flask and HTML allows users to interact with the sentiment analysis tool, providing real-time feedback.

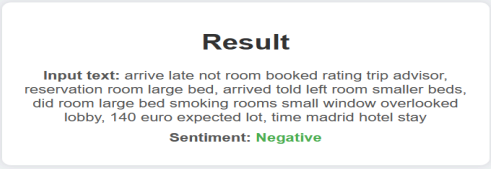
All the data source code and documents have been pushed into git

Conclusion

Our sentiment analysis model offers a scalable and efficient solution for classifying customer reviews in the hospitality industry. By accurately categorizing feedback, businesses can identify areas for improvement and enhance customer satisfaction.


Sentiment Analysis
Enter your text:
Type your text here...
SUBMIT


Sentiment Analysis
Enter your text:
arrive late not room booked rating trip advisor, reservation room large bed, arrived told left room smaller beds, did room large bed smoking rooms small window overlooked lobby, 140 euro expected lot, time madrid hotel stay
SUBMIT


Result
Input text: arrive late not room booked rating trip advisor, reservation room large bed, arrived told left room smaller beds, did room large bed smoking rooms small window overlooked lobby, 140 euro expected lot, time madrid hotel stay
Sentiment: Negative