NAAN MUDHALVAN

ARTIFICIAL INTELLIGENCE

PROJECT TITLE

SENTIMENT ANALYSIS FOR MARKETING

REG NO: 7122211016
NAME: PRIYADARSHINI.S
DEPT: COMPUTER SCIENCE & ENGINEERING
YEAR & SEM: 3rd & 5th
COLLEGE NAME: PARK COLLEGE OF ENGINEERING AND TECHNOLOGY

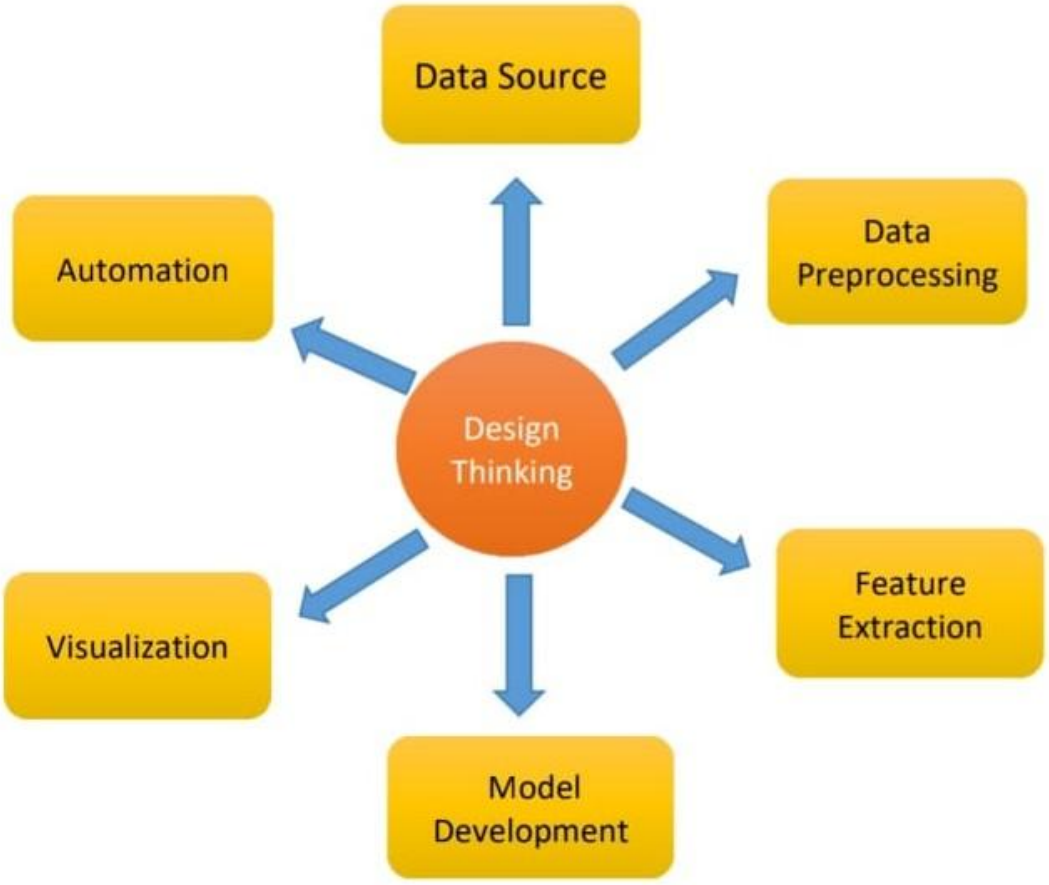
PHASE 1

PROBLEM DEFINITION AND DESIGN THINKING

PROBLEM DEFINITION

Sentiment analysis, also referred to as opinion mining, is an approach to natural language processing (NLP) that identifies the emotional tone behind a body of text. This is a popular way for organizations to determine and categorize opinions about a product, service or idea.

DESIGN THINKING



PHASE 2

PROJECT INNOVATION IDEA

STEPS:

1. Data source
2. Sentiment analysis tool
3. Collect data
4. Preprocess the data
5. Perform sentiment analysis
6. Monitor and sentiment

PHASE 3

DATA ANALYSIS AND PREPROCESSING

STEP 1:

Data Collection:

- Gather data from various sources, such as social media, customer reviews, surveys, or other text-based sources.
- Ensure that the data represents the target audience or market segment you are interested in.

STEP 2:

Data Cleaning:

- Remove any irrelevant or redundant information from the text, such as URLs, special characters, or numbers.
- Handle text encoding issues if present.
- Address issues like misspellings and abbreviations to ensure consistency.

STEP 3:

Removing Noise:

- Identify and remove irrelevant or noisy text elements that may not carry sentiment information, such as HTML tags, mentions, or hashtags.

STEP 4:

Handling Negations:

- Identify and mark negations in the text to change the sentiment of the following words. For example, "not good" should be interpreted as a negative sentiment.

STEP 5:

Sentiment Labeling:

- Manually or automatically label the data with sentiment labels (e.g., positive, negative, neutral) for supervised learning.
- Consider using pre-trained sentiment lexicons to help with sentiment labeling.

STEP 6:

Data Visualization:

- Create visualizations like word clouds, histograms, or bar charts to gain insights into the data and sentiment distribution.

PHASE 4

TITTLE: selecting a machine learning language algorithm,training the model,and evaluating its performance.

MACHINE LANGUGAE FOR SENTIMENT ANALYSIS:

The supervised machine learning technique best suits sentiment analysis because it can train large data sets and provide robust results. It is preferable to semi-supervised and unsupervised methods because it relies on data labeled manually by humans so includes fewer errors.

Conclusion:

Evaluating and selecting machine learning algorithms is a crucial step in building successful predictive models. By understanding the types of algorithms, defining evaluation criteria, preparing the dataset, implementing the algorithms, and employing appropriate evaluation techniques, you can make informed decisions. Remember that the iterative process of model tuning and fine-tuning is essential to achieve optimal results. By following these guidelines, you can leverage the power of machine learning algorithms to drive accurate predictions and unlock valuable insights in your domain.

Training the model of sentiment analysis:

To train the sentiment classifier, convert the words to word vectors using the pretrained word embedding emb. First remove the words that do not appear in the word embedding emb.

```
idx = ~isVocabularyWord(emb,data.Word);
data(idx,:) = [];
```

Set aside 10% of the words at random for testing.

```
numWords = size(data,1);
cvp = cvpartition(numWords,'HoldOut',0.1);
dataTrain = data(training(cvp),:);
dataTest = data(test(cvp),:);
```

Convert the words in the training data to word vectors using word2vec.

```
wordsTrain = dataTrain.Word;
XTrain = word2vec(emb,wordsTrain);
YTrain = dataTrain.Label;
```

Train Sentiment Classifier

Train a support vector machine (SVM) classifier which classifies word vectors into positive and negative categories.

```
mdl = fitcsvm(XTrain,YTrain);
```

Test Classifier

Convert the words in the test data to word vectors using word2vec.

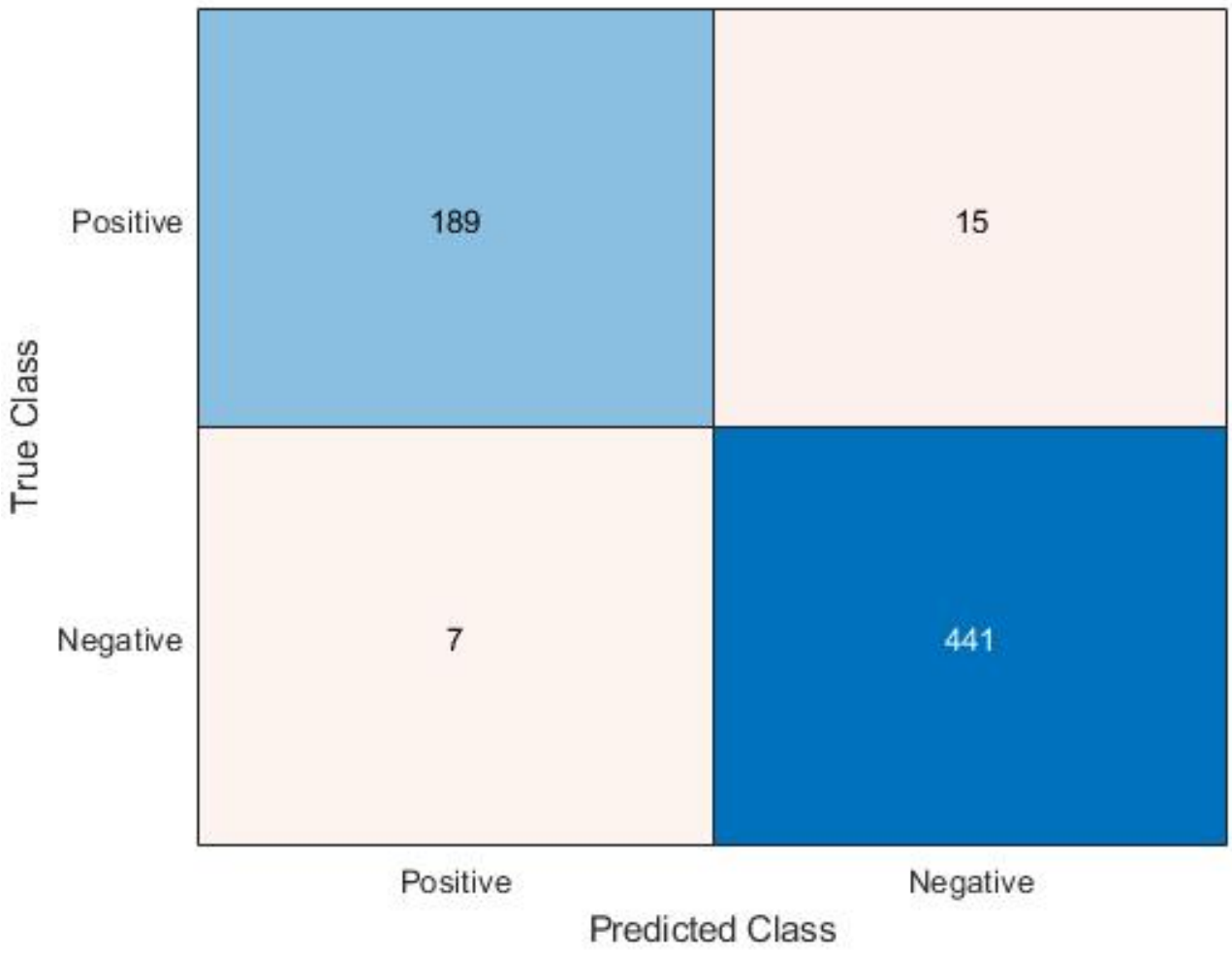
```
wordsTest = dataTest.Word;
XTest = word2vec(emb,wordsTest);
YTest = dataTest.Label;
```

Predict the sentiment labels of the test word vectors.

```
[YPred,scores] = predict(mdl,XTest);
```

Visualize the classification accuracy in a confusion matrix.

```
figure
confusionchart(YTest,YPred);
```



Visualize the classifications in word clouds. Plot the words with positive and negative sentiments in word clouds with word sizes corresponding to the prediction scores.

```
subplot(1,2,1)
idx = YPred == "Positive";
wordcloud(wordsTest(idx),scores(idx,1));
title("Predicted Positive Sentiment")

subplot(1,2,2)
wordcloud(wordsTest(~idx),scores(~idx,2));
title("Predicted Negative Sentiment")
```

Predicted Positive Sentiment

Predicted Negative Sentiment

Calculate Sentiment of Collections of Text

To calculate the sentiment of a piece of text, for example an update on social media, predict the sentiment score of each word in the text and take the mean sentiment score.

```
filename = "weekendUpdates.xlsx";
tbl = readtable(filename,'TextType','string');
textData = tbl.TextData;
textData(1:10)
```

```
ans = 10×1 string array
   "Happy anniversary! ❤ Next stop: Paris! ✈ #vacation"
   "Haha, BBQ on the beach, engage smug mode! ❤     #vacation"
   "getting ready for Saturday night     #yum #weekend    "
   "Say it with me - I NEED A #VACATION!!! ☹  "
   "    Chilling     at home for the first time in ages…This is the life!     #weekend"
   "My last #weekend before the exam       ."
   "can’t believe my #vacation is over     so unfair"
   "Can’t wait for tennis this #weekend            "
   "I had so much fun!          Best trip EVER!          #vacation #weekend"
   "Hot weather and air con broke in car     #sweaty #roadtrip #vacation"
```