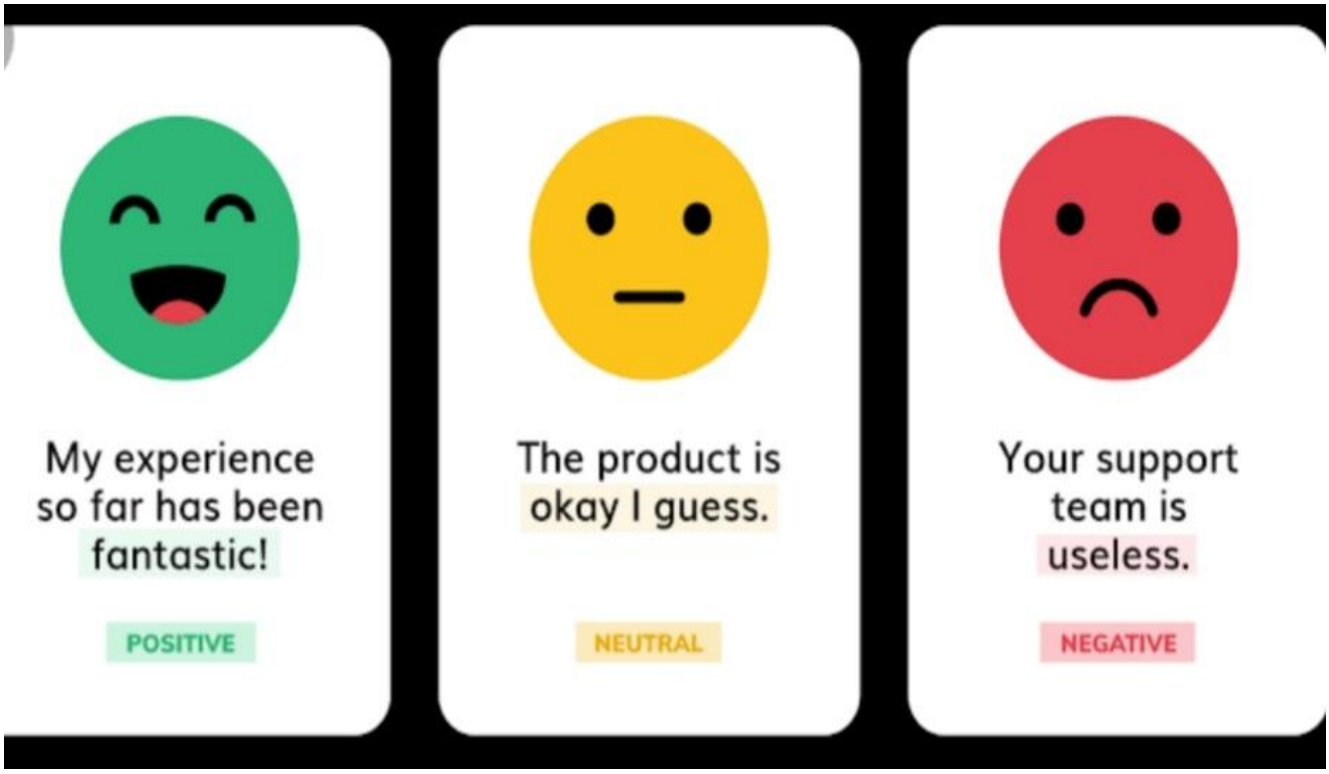
SENTIMENT ANALYSIS IN MARKETING USING MACHINE LEARNING.

PROJECT TiTle : SENTIMENT ANALYSIS IN MARKETING.

Phase 4: Development Part-2.

Topic :Continue developing the sentiment analysis model by feature engineering,model training and evaluation.



SENTIMENT ANALYSIS IN MARKETING.

Introduction:

- With advancements in technology and fields like deep learning, sentiment analysis is becoming more and more common for companies that want to gauge their customers' sentiments.
- Today, businesses use natural language processing, statistical analysis, and text analysis to identify the sentiment and classify words into positive, negative, and neutral categories.
- The best companies understand the importance of understanding their customers' sentiments – what they are saying, what they mean and how they are saying. You can use xsentiment analysis to identify customer sentiment in comments, reviews, tweets, or social media platforms where people mention your brand.
- As sentiment analysis is the domain of understanding emotions using software, we have prepared a complete guide to understand 'what is sentiment analysis?', its tools, and different classifications and use cases.
- Sentiment analysis can be defined as analyzing the positive or negative sentiment of the customer in text. The contextual analysis of identifying information helps businesses understand their customers' social sentiment by monitoring online conversations.

The sentiment analysis process mainly focuses on polarity, i.e., positive, negative, or neutral. Apart from polarity, it also considers the feelings and emotions(happy, sad, angry, etc.), intentions(interested or not interested), or urgency(urgent or not urgent) of the text.

Overview of the process:

- ✓ Prepare the data.
- ✓ Perform feature selection.
- ✓ Train the model.
- ✓ Evaluate the model.
- ✓ Deploy the model.

OVERVIEW OF THE PROCESS:

The following is an overview of the process of building a house price prediction model by feature selection, model training, and evaluation:

1.Prepare the data: This includes cleaning the data, removing outliers, and handling missing values.

2.*Perform feature selection:* This can be done using a variety of methods, such as correlation analysis, information gain, and recursive feature elimination.

3.Train the model: There are many different machine learning algorithms that can be used for house price prediction. Some popular choices include linear regression, random forests, and gradient boosting machines.

4.Evaluate the model: This can be done by calculating the mean squared error (MSE) or the root mean squared error (RMSE) of the model's predictions on the held-out test set.

5.Deploy the model: Once the model has been evaluated and found to be performing well, it can be deployed to production so that it can be used to predict the house prices of new houses.

PROCEDURE:

1. **Identify the target variable**. This is the variable that you want to predict, such as house price.

2. **Explore the data**. This will help you to understand the relationships between the different features and the target variable. You can use data visualization and correlation analysis to identify feature that are highly correlated with the target variable.

3. **Remove redundant features**. If two features are highly correlated with each other, then you can remove one of the features, as they are likely to contain redundant information.

4. **Remove irrelevant features**. If a feature is not correlated with the target variable, then you can remove it, as it is unlikely to be useful for prediction.

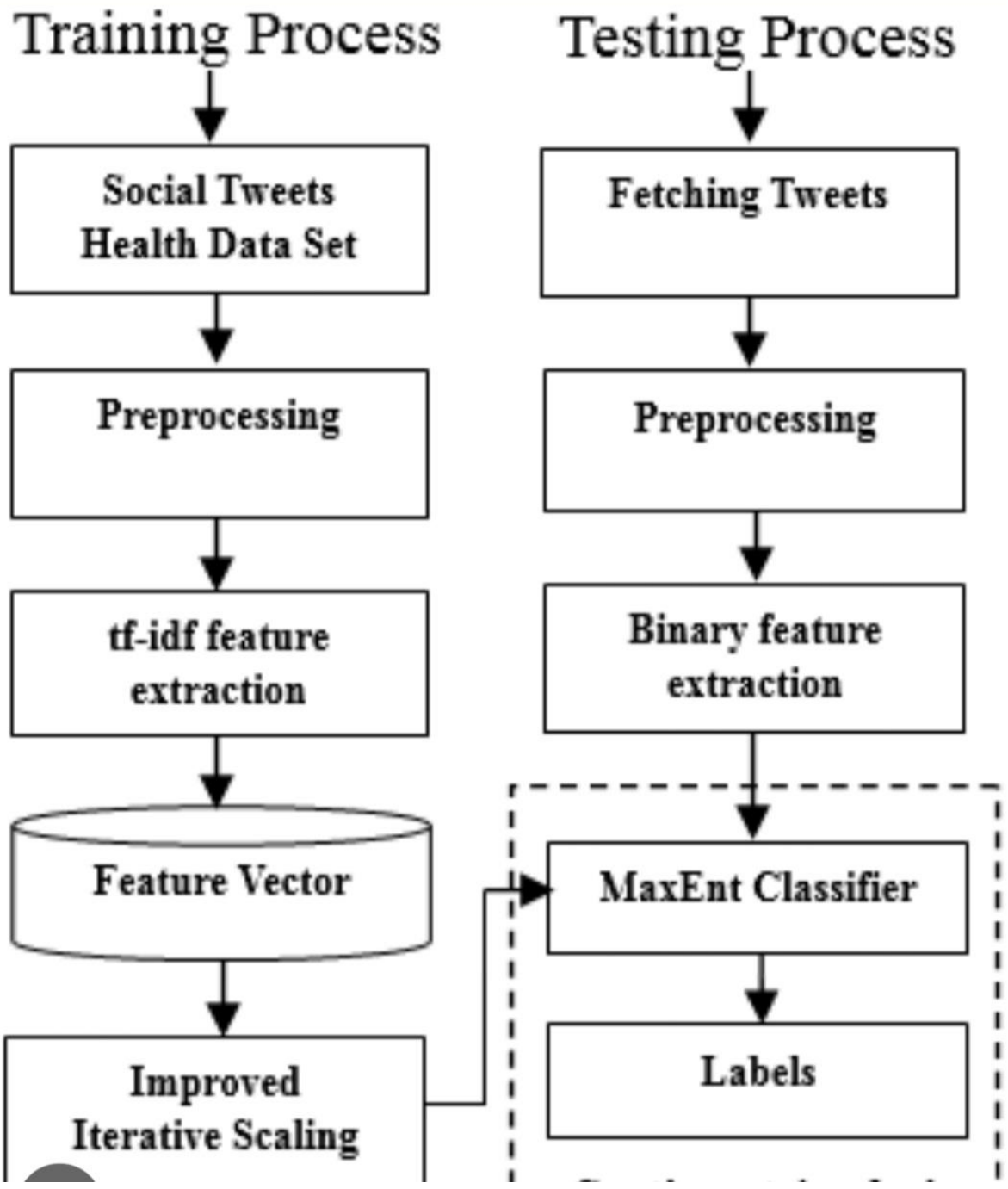
FEATURE SELECTION:

A list of possible features and their variations was produced by analyzing available literature. The following features were selected for testing, since they proved useful in previous research The different types of features used were unigram bag-of-words,bigram bag-of-words, special characters counts, and other

textual properties. The motivation of using these is presented in this section.

Unigram bag of words:

The bag-of-words feature is a word counter since it is implemented in the unigram case. Each feature in the input vector represents the count of a certain word in the input sentence. Stemming and lower casing was used to decrease the number of unique words, and join different inflections of the same primitive together. However, the information loss of removing the end of words must be taken into consideration.


Training Process
Testing Process
Social Tweets Health Data Set
Fetching Tweets
Preprocessing
Preprocessing
tf-idf feature extraction
Binary feature extraction
Feature Vector
MaxEnt Classifier
Improved Iterative Scaling
Labels

Model training:

```
>>> import pandas as pd
>>> df =
pd.read_csv('/datasets/sentiment/amazon_cells_labelled.txt',
names=['review', 'sentiment'], sep='\t')
>>> df.head()0 So there is no way for me to plug it in here i… 0
1 Good case, Excellent value. 1
2 Great for the jawbone. 1
3 Tied to charger for conversations lasting more… 0
4 The mic is great.
```

Splitting the Data Set into a Training Set and aTest Set:

```
>>> from sklearn.model_selection import train_test_split
>>> reviews = df['review'].values
>>> labels = df['sentiment'].values
>>> reviews_train, reviews_test, y_train, y_test =
train_test_split(reviews, labels, test_size=0.2, random_state=1000)
```

Transforming Text into Numerical Feature Vectors:

```
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> vectorizer = CountVectorizer(tokenizer = spacy_tokenizer,
ngram_range=(1,1))
>>> #By default, the vectorizer might be created as follows:
>>> #vectorizer = CountVectorizer()
>>> vectorizer.fit(reviews_train)
```

Training the Model:

```
>>> from sklearn.linear_model import LogisticRegression
>>> classifier = LogisticRegression()
>>> classifier.fit(X_train, y_train)
```

Evaluation of the Model:

```
>>> accuracy = classifier.score(X_test, y_test)
>>> print("Accuracy:", accuracy)Accuracy: 0.785
```

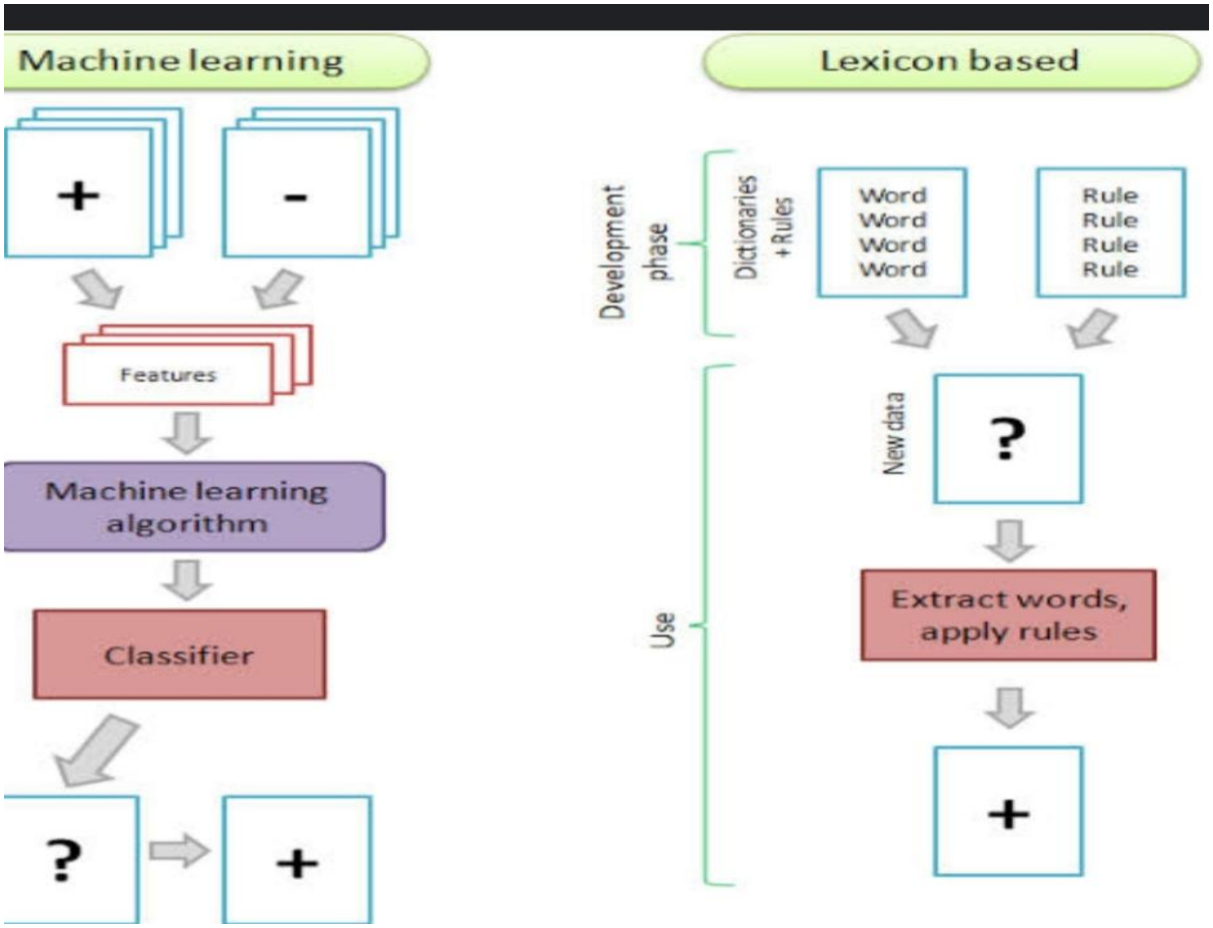
Predictions on New Data:

```
>>> new_reviews = ['Old version of python useless', 'Very good
effort, but not five stars', 'Clear and concise']
>>> X_new = vectorizer.transform(new_reviews)
>>> classifier.predict(X_new)array([0, 1, 1])
```

MODEL EVALUATION :

Consumer feedback is highly valuable in business to assess their performance and is also beneficial to customers as it gives them an idea of what to expect from new products. In this research, the aim is to evaluate different deep learning approaches to accurately predict the opinion of customers based on mobile phone reviews . The prediction is based on analysing these reviews and categorizing them as positive, negative, or neutral. Different deep learning algorithms have been implemented and evaluated such as simple RNN with its four variants, namely, Long Short-Term Memory Networks (LRNN), Group Long Short-Term Memory Networks (GLRNN), gated recurrent unit (GRNN), and update recurrent unit (UGRNN). All evaluated algorithms are combined with word embedding as feature extraction approach for sentiment analysis including Glove, word2vec, and FastText by Skip-grams. The five different algorithms with the three feature extraction methods are evaluated based on accuracy, recall, precision, and F1-score for both balanced and unbalanced datasets. For the unbalanced dataset, it was found that the GLRNN algorithms with FastText feature extraction scored the highest accuracy of 93.75%. This result achieved the highest accuracy on this dataset when compared with other methods mentioned in the literature. For the balanced dataset, the highest achieved accuracy was 88.39% by the LRNN algorithm..

FEATURE TO PERFORM MODEL TRAINING:



Feature engineering is the process of transforming raw data into features that are more informative and predictive for machine learning models. By using a variety of feature engineering techniques, you can create a set of features that will help your model to predicts more accurately.Use

cross-validation.Cross-validation is a technique for evaluating the performance of a

machine learning model on unseen data. It is important to use cross- validation to evaluate the performance of your model during the evaluation process. This will help you to avoid overfitting and to ensure that the model will generalize well to new data.

- Use cross validation.
- Use ensemble methods.
- Compare model to a damper.
- Use a hold out test set.
- Analysis model for prediction.

  Use cross-validat:ion.

  Cross-validation is a technique for evaluating the performance of a machine learning model on unseen data. It is important to use cross-validation to

evaluate the performance of your model during the trainingprocess. This will help you to avoid overfitting and to ensure that your model will generalize well to new data.

Use ensemble methods:

Ensemble methods are machine learning methods that combine the predictions of multiple models to produce a more accurate prediction.Ensemble methods can often achieve better performance than individual machine learning models.

Use cross-validation.

Cross-validation is a technique for evaluating the performance of a machine learning model on unseen data. It is important to use cross- validation to evaluate the performance of your model during the

evaluation process. This will help you to avoid overfitting and to ensure that the model will generalize well to new data.

Use a **holdout test set**.

A holdout test set is a set of data that is not used to train or evaluate the model during the training process. This data is used toevaluate the performance of the model on unseen data after the training process is complete.

Compare the model to a baseline.

A baseline is a simple model that is used to compare the performance of your model to. For example, you could use the means a baseline.

Analyze the model's predictions.

Once you have evaluated the performance of the model, you can analyze the model's predictions to

identify any patterns or biases. Thiswill help you to understand the strengths and weaknesses of the model and to improve it