SENTIMENT ANALYSIS FOR MARKETING

BATCH MEMBER

961621205023 : vyshnavi B S

Phase 3 submission document

Project Title: sentiment analysis for marketing

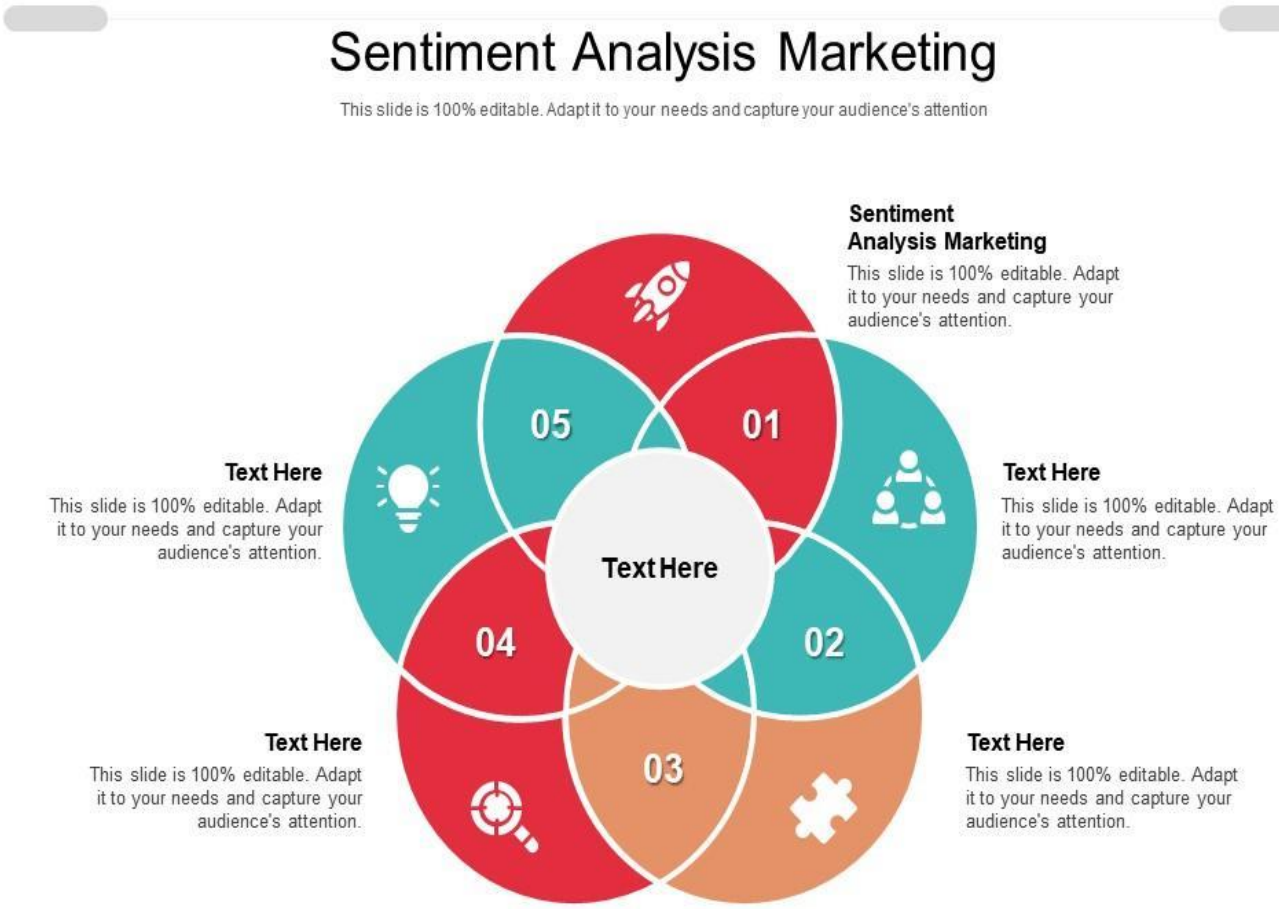
Phase 3: Development Part 1

Sentiment Analysis Marketing

Parsing online feedback can be a challenging task for any business with a strong social presence. In marketing, sentiment analysis can be useful for teams that want to examine commentary about their brands from a qualitative angle.

By looking at the tone and content of social sentiments, you can develop enhanced metrics that may offer more valuable insight.

In this article, we define sentiment analysis in marketing, describe its benefits and offer tips to help you incorporate it into your team’s marketing plan.



What is sentiment analysis in marketing?

Sentiment analysis is a marketing tool that helps you examine the way people interact with a brand online. This method is more comprehensive than traditional online marketing tracking, which measures the number of online interactions that customers have with a brand, like comments and shares.

Using sentiment analysis, you can label individual interactions as positive, negative or neutral. Once you've figured out how to determine and track these labels, you can use this new data set for a variety of marketing purposes, including your online strategy.

Sentimental analysis is an extremely useful tool to have since higher numbers of interactions don't always equate to better results.

For example, if you were to receive 10 replies on a social post and all of them were positive, your post likely had a more compelling effect on your audience than if you receive 100 replies with only 10 of them being positive.

The primary purpose of sentiment analysis is to respond to commentary more constructively.

3 types of sentiment analysis

To perform sentiment analysis, a marketing team might use a software platform that creates an algorithm to monitor customer engagement online. There are three fundamental ways to develop an algorithm for distinguishing social sentiment:

1. **Manual analysis:** This type uses manually created rules based on neurolinguistic principles, such as stemming and tokenization. It takes a long time to set up, but it's easy to change and customize.
2. **Automatic analysis:** This type uses machine learning techniques that use neural networks and statistical models to classify language. It can be challenging to change, but it's easy to set up and manage.
3. **Hybrid analysis:** This type uses both rules-based and machine-learning analyses. It's a balanced approach that most social listening applications employ.

What are the benefits of sentiment analysis?

Most online platforms implement their own algorithms to display content, so sentimental analysis allows you to track relevant commentary in a way that's most useful for your purposes. Conversations about a brand y can begin and end quickly, so using a standardized method to track them can give you valuable insights into customer browsing habits. Here are some of the benefits of sentiment analysis:

Understanding your audience and defining your niche:

Sentiment analysis allows you to look at your audience from a much more granular perspective, which can help you identify a market niche that fits the products and services your company sells. Understanding what your brand means to your current customers can help you increase your brand’s market share much more quickly. For example, the owners of a struggling ice cream parlor can benefit from learning which flavors people like or dislike. They might improve profits by marketing liked flavors more heavily, and discontinuing less popular flavors could help reduce their operating costs.

Improving customer service support and managing PR issues

Many businesses use social media channels for customer service support because it lets them resolve issues in a personalized yet immediate way. Responding to negative comments can help de-escalate situations before they grow into something less manageable. For example, if a customer were to tag your brand in a post in which they're upset about a defective product, you can respond publicly to apologize. You can then follow up by messaging them privately to reinforce your commitment to quality. Handling negative sentiments effectively and publicly can also show other customers that the company has excellent service policies.

Adjusting messaging and product development

Sentiment analysis is an inexpensive way to improve messaging and product development. Knowing what customers value about a product or service can tell you what to emphasize in your promotional material. For example, if there's been a sudden and unexplained spike in sales of a certain product, you can check your positive mentions to see what customers are saying. You may find one of your products has suddenly become popular when someone posted about a feature that isn't in your other products.

Monitoring competitors

Both negative and positive mentions can sometimes include valuable information about your brand's competitors. Seeing how customers view your product and services compared to others can help you create promotional strategies that emphasize your brand's advantages. For example, customers on social media might mention that they prefer the coupons or discount codes that a competing brand offers. This information can help the marketing team revise promotional material with more competitive sales offers.

Identifying influencers

Using a sentiment analysis system can help marketers identify micro-influencers, who are social media figures with a relatively small number of highly engaged followers. Typically, micro-influencers are the ones who are the most helpful to your brand, as they have more direct interaction with their followers. For example, you might discover that one of your customers listens to a podcast with about 10,000 subscribers and the audience overlaps with your own. You may then want to reach out to the creator to see if they're interested in a sponsorship.

Tips for including sentiment analysis in your marketing

Marketing professionals can use sentiment analysis in many ways, so it's important to know how to use it to the fullest extent possible. Here are some tips for how to use sentiment analysis in marketing:

Create specific goals and benchmarks for reaching them

Before you choose an analysis method and start identifying positive, neutral and negative sentiments, set specific goals for improving customer sentiment. The behavior you want to see depends on your aims and goals. Some companies may value positive mentions over amount, while others want to reduce neutral social sentiment. Understanding your goals can help you to design an algorithm that produces reports with consistent indicators for success.

Measure data before and after campaigns

It's important to track data prior to implementing a campaign based on sentiment analysis, so you can have something to compare. Comparing prior data with the results of a social media sentiment campaign can tell a marketer whether their methods are effective. The data you're gathering is unique to your company's brand and consumer base, so you might conduct customer research to evaluate current sentiment about the company or brand. Often, marketing teams contract with consumer research consultants, who gather information from a wide range of sources and provide in-depth reports of current customer sentiment.

Consider using a sentiment analysis program

Shifts in social sentiment can happen fast, so the ability to react immediately is vital. Using a software program specifically built to monitor sentiment in real time can allow marketers to respond quickly to negative or positive comments. These tools often come with pre-programmed algorithms, which make them ideal for marketing teams without an in-house programmer. They might display sentiment reports on interactive dashboards or send users notifications if they detect an interaction with a strong sentiment.

Understand the limitations of sentiment analysis tools

Algorithms can be sophisticated enough to recognize sarcasm, but there are still limitations to how much sentiment analysis tools can do. It's a good idea to spot-check some of the data yourself to ensure the accuracy of how it's being interpreted by these tools. Certain language elements are especially difficult for neural networks to decipher, such as slang, irony and figurative language. You might distribute feedback surveys or collect testimonials from customers to supplement the information you get from an analysis tool. These resources can help you interpret the results of a sentiment analysis.

Sentiment Analysis using Python [with source code]

Sentiment Analysis – One of the most popular projects in the industry. Every customer facing industry (retail, telecom, finance, etc.) is interested in identifying their customers' sentiment, whether they think positive or negative about them.
Python sentiment analysis is a methodology for analyzing a piece of text to discover the sentiment hidden within it. It accomplishes this by combining machine learning and natural language processing (NLP). Sentiment analysis allows you to examine the feelings expressed in a piece of text.

When we have everything in place, we can start by importing everything we will need during implementation.

```
import torch
from torch import nn
from torch import optim
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from torchtext.data.utils import get_tokenizer
from torchtext.vocab import build_vocab_from_iterator
from torch.utils.data import DataLoader, Dataset
```

Additionally, we have to specify what device we will be training an LSTM on:

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

I strongly recommend using **cuda**. Otherwise, training will be very time-consuming. If you don't have access to a GPU machine, consider using a free one on collab or kaggle.

And as the last setup step, we'll specify a global random seed for reproducibility:

```
np.random.seed(42)
```

Finally, we will define a path to our dataset:

```
DATA_PATH = "./trip-advisor-hotel-reviews/tripadvisor_hotel_reviews.csv"
```

Loading and Inspecting Data

First things first, loading data to panda's **DataFrame**:

```
df = pd.read_csv(DATA_PATH)
df.head()
```

n a real-world situation, this `Rating` column would be more than enough for us to train the review scoring regression model. But instead of that, we'll demonstrate how to work with less informative data. We'll categorize reviews into three categories:

- negative: ratings below 4
- neutral: ratings equal to 4
- positive: ratings equal to 5

```
neutral_range = {"low": 4, "high": 5}
df["Sentiment"] = "neutral"
df["Sentiment"].loc[df["Rating"] < neutral_range["low"]] = "negative"
```

```
df["Sentiment"].loc[df["Rating"] >= neutral_range["high"]] = "positive"
df.head()
```

Data split and Baseline

As usual, we'll be splitting our data into train and validation subsets while ensuring that the resulting split is stratified.

```
X_train, X_validation, y_train, y_validation = train_test_split(df["Review"], df["Sentiment"], test_size=0.2,
stratify=df["Sentiment"])
```

To correctly evaluate created model, we have to establish some initial baseline first. In our case (sentiment classification), we can build a simple model that picks sentiment at random.

```
class RandomBaseline:


def __init__(self):
self.categories = {}


def fit(self, data, target_col):
cat_names = data[target_col].unique()
agg = data.groupby(target_col).count()
for n in cat_names:
self.categories[n] = agg.loc[n][0] / len(data)


def predict(self, data):
return np.random.choice(list(self.categories.keys()), len(data), list(self.categories.values()))
```

Now we can "train" our random model.

```
rb = RandomBaseline()
rb.fit(df.iloc[X_train.index], "Sentiment")
pred = rb.predict(X_test)
accuracy_score(y_test, pred)
```

```
# 0.3273969260795316
```

As expected, accuracy for randomly picking 1 from 3 categories is close to 33%

Data preparation

Before we start modeling, we have to transform reviews to form “understandable” for the neural network. We’ll do it by:

1. tokenizing each review – converting it to a list of words (tokens)
2. building a vocabulary – mapping each token to index

tokenization:

```
tokenizer = get_tokenizer('basic_english')
tokenizer("the place was nice")
# ['the', 'place', 'was', 'nice']
```

building a vocabulary:

```
def tokenized_review_iterator(reviews):
for r in reviews:
yield tokenizer(r)


vocab = build_vocab_from_iterator(tokenized_review_iterator(X_train), special
s=["<unk>"])
vocab.set_default_index(vocab["<unk>"])
vocab(['the', 'place', 'was', 'nice'])
# [33, 31, 3826, 15]
```

Now we need to implement a custom PyTorch Dataset that will handle preparing and serving data during training and evaluation.

```
target_map = {
"positive": 0,
"neutral": 1,
"negative": 2
}

text_pipeline = lambda x: vocab(tokenizer(x))
label_pipeline = lambda x: target_map[x]

class ReviewDataset(Dataset):

def __init__(self, X, y, text_pipeline, label_pipeline):
self.X = X
self.y = y
self.text_pipeline = text_pipeline
self.label_pipeline = label_pipeline

def __len__(self):
return len(self.X)

def __getitem__(self, idx):
text = torch.tensor(self.text_pipeline(self.X.iloc[idx]))
length = torch.tensor(len(text))
label = torch.tensor(self.label_pipeline(self.y.iloc[idx]))
return {"text": text, "length": length, "labels": label}

train_dataset = ReviewDataset(X_train, y_train, text_pipeline, label_pipeline
)
test_dataset = ReviewDataset(X_test, y_test, text_pipeline, label_pipeline)
```

Additionally, as we work with reviews that don't have equal lengths, we'll have to provide a function that will pad shorter sequences in batch with blank tokens.

```
def collate(batch):
```

```
batch.sort(key=lambda x: x["length"], reverse=True)
text, lengths, labels = zip(*[d.values() for d in batch])
text = torch.nn.utils.rnn.pad_sequence(text, batch_first=True)
lengths = torch.stack(lengths)
labels = torch.stack(labels)
return text, lengths, lab
```