

The food was great! But I didn't like the service.

I will definitely come again. Great menu.

The atmosphere is nice, and the service was helpful.  
When it comes to food, I would say 7/10.

Not my style, I don't recommend it.

## Sentiment Analysis in Python

Textblob vs Vader Sentiment vs Flair vs Building It From Scratch



# What is sentiment analysis?

Sentiment analysis is the task of determining the emotional value of a given expression in natural language.

It is essentially a multiclass text classification task where the given input text is classified into positive, neutral, or negative sentiment. The number of classes can vary according to the nature of the training dataset.


Application of sentiment analysis

- Movie reviews: Analysing online movie reviews to get insights from the audience about the movie.
- News sentiment analysis: analyzing news sentiments for a particular organization to get insights.

# 1. Rule based sentiment analysis

- Textblob
- VADER

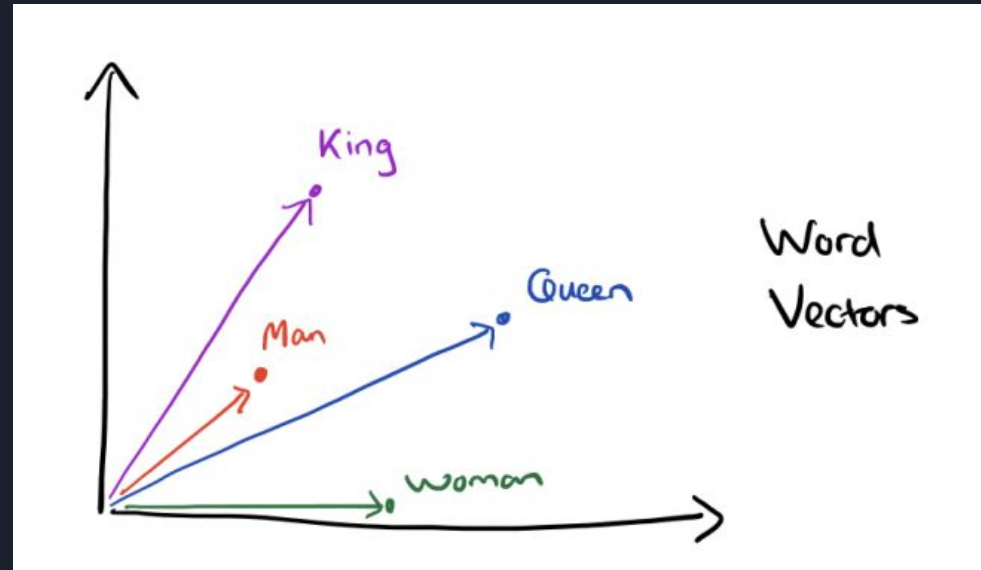
Sentiment Analysis

 <p>My experience so far has been fantastic!</p> <p>POSITIVE</p>	 <p>The product is ok I guess</p> <p>NEUTRAL</p>	 <p>Your support team is useless</p> <p>NEGATIVE</p>
--	---	---

 MonkeyLearn

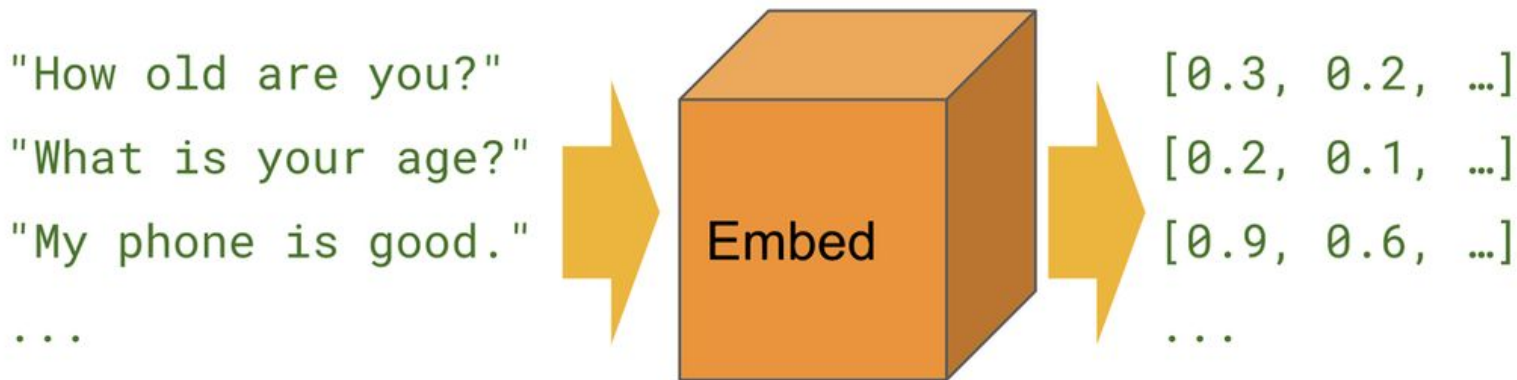
## 2. Embedding Based sentiment analysis

- Flair
- Custom Model



# How to represent sentences?

## Universal sentence encoder





# Let's get our hands dirty with some code !

- Download your kaggle API json
- Open the notebook

[bit.ly/3iJajEZ](https://bit.ly/3iJajEZ)



**Mljar-supervised**

**<https://github.com/mljar/mljar-supervised>**

**mljar**

**Machine Learning for Humans**

# No-code ML

## 👉 Binary Classification Example

There is a simple interface available with `fit` and `predict` methods.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from supervised.automl import AutoML

df = pd.read_csv(
    "https://raw.githubusercontent.com/pplonski/datasets-for-start/master/adult/data.csv",
    skipinitialspace=True,
)
X_train, X_test, y_train, y_test = train_test_split(
    df[df.columns[:-1]], df["income"], test_size=0.25
)

automl = AutoML()
automl.fit(X_train, y_train)

predictions = automl.predict(X_test)
```





# ASK ME ANYTHING!!

