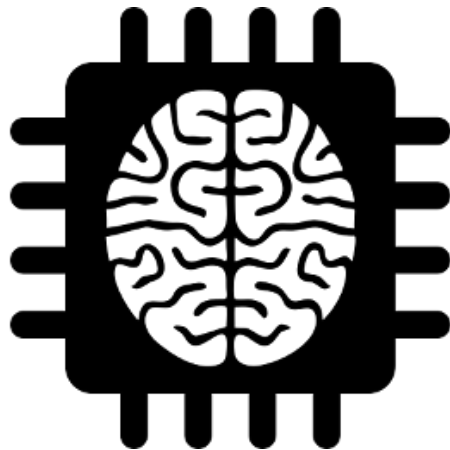


# Análise de sentimentos de avaliações de aplicativos da Google Play Store usando AWS Comprehend

# Relevância

- Porque o problema é interessante?
- O que se tem a ganhar para solucioná-lo?
- Qual sua relação com aprendizado de máquina?

- Pesquisa em PLN.
- Fácil de ser implementado.
- Facilitador no relacionamento Desenvolvedor - Usuário.
- Utiliza algoritmos de AM.



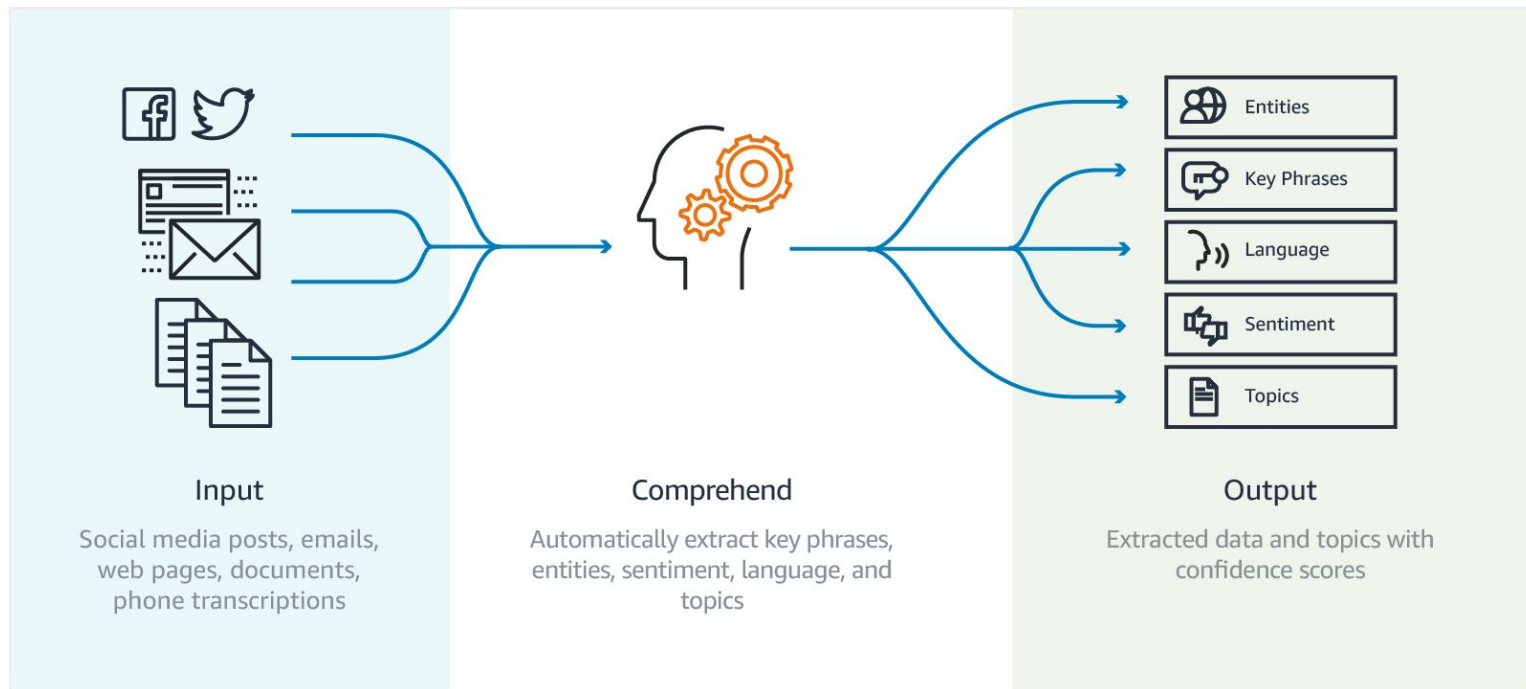
# Metodologia

- Metodologia utilizada?
- Técnicas aplicadas?

- Python.
- Google Play Store.
- AWS Comprehend.



# Amazon Comprehend



# Demo

```
from dateutil.parser import parse
```

```
reload(sys)
sys.setdefaultencoding('utf8')
```

```
def get_app_coments(pagenum, appid):
    print("Baixando comentarios...")
    url = "https://play.google.com/store/getreviews"

    headers = {
        "Content-Type": "application/x-www-form-urlencoded;charset=ascii"
    }

    payload = "reviewType=0&pageNum=" + pagenum + "&id=" + appid + "&reviewSortOrder=2&xhr=1&hl"

    text = ast.literal_eval(requests.post(url, data=payload, headers=headers).text.replace("\n"))
    text = text.replace("\u003c", "<").replace("\u003e", ">").replace("\u003a", ":").replace("\u0026", "&")

    soup = BeautifulSoup(text, "lxml")
```

```
    dates = soup.findAll("span", { "class" : "review-date" })
    reviews = soup.findAll("div", { "class" : "review-body with-review-wrapper" })
```

```
    result = []
```

```
    for i in range(len(dates)):
        result.append([dates[i].contents[0], reviews[i].contents[2].encode("utf-8")])
```

```
    print("Comentarios baixados.")
    return result
```

```
def get_sentiment(reviews):
    comprehend = boto3.client(service_name="comprehend")
```

```
    print("Analisando sentimentos...")
    for review in tqdm(reviews):
        review[1] = comprehend.detect_sentiment(Text=review[1], LanguageCode="en")["SentimentScore"]

    return reviews
```

```
def merge_equal_days(reviews):
```

```
    for i in range(len(reviews)-1):
        if reviews[i] != None:
            aux = reviews[i][1]
            divisor = 1
            for j in range(i+1, len(reviews)):
                if reviews[j] != None and reviews[i][0] == reviews[j][0]:
                    aux = sum([aux, reviews[j][1]])
```

# Resultados

- Resultados casuais?
- Conclusões?
- Trabalhos futuros?

- Resultados fiéis à avaliação humana.
- Facilidade de implementação.
- Baixo custo.
- Utilização de mais recursos disponíveis na AWS.
- Implementação de algoritmos.

