***Problem statement - There are times when a user writes Good, Nice App or any other positive text, in the review and gives 1-star rating. Your goal is to identify the reviews where the semantics of review text does not match rating.***

# Importing Libraries

In [14]:
```
import pandas as pd
import numpy as np
```

In [15]:
```
import nltk
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\ABHIRAM\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

Out[15]: True

In [16]:
```
# importing data

df=pd.read_csv("chrome_reviews.csv")
```

In [17]:
```
df.head()
```

Out[17]:

| | ID | Review URL | Text | Star | Thumbs Up | User Name | Developer Reply | |
|---|---|---|---|---|---|---|---|---|
| **0** | 3886 | https://play.google.com/store/apps/details?id=... | This is very helpfull aap. | 5 | 0 | INDIAN Knowledge | NaN | 83 |
| **1** | 3887 | https://play.google.com/store/apps/details?id=... | Good | 3 | 2 | Ijeoma Happiness | NaN | 85 |
| **2** | 3888 | https://play.google.com/store/apps/details?id=... | Not able to update. Neither able to uninstall. | 1 | 0 | Priti D BtCFs-29 | NaN | 85 |
| **3** | 3889 | https://play.google.com/store/apps/details?id=... | Nice app | 4 | 0 | Ajeet Raja | NaN | 77 |
| **4** | 3890 | https://play.google.com/store/apps/details?id=... | Many unwanted ads | 1 | 0 | Rams Mp | NaN | 8 |

In [18]:
```
df.shape
```

Out[18]: (7204, 10)

In [19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7204 entries, 0 to 7203
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   ID               7204 non-null   int64 
 1   Review URL       7204 non-null   object
 2   Text             7203 non-null   object
 3   Star             7204 non-null   int64 
 4   Thumbs Up        7204 non-null   int64 
 5   User Name        7204 non-null   object
 6   Developer Reply  95 non-null     object
 7   Version          7119 non-null   object
 8   Review Date      7204 non-null   object
 9   App ID           7204 non-null   object
dtypes: int64(3), object(7)
memory usage: 562.9+ KB
```

## *Text Preprocessing*

In [20]:
```
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer



class Text_Preprocessor:
    """It will preprocess the given text data.
       Written by : Vikram Singh
       Date: 05/01/2022"""

    def __init__(self):
        pass

    def text_cleaner(self, data):
        """Method Name: text_cleaner
           Description: It will do all the basic text cleaning steps & return cle
        try:
            ps = PorterStemmer()
            cleaned_data = re.sub('[^a-zA-Z]', ' ', data)
            cleaned_data = cleaned_data.lower()
            cleaned_data = cleaned_data.split()
            cleaned_data = [ps.stem(word) for word in cleaned_data if not word ir
            cleaned_data = ' '.join(cleaned_data)
            return cleaned_data
        except Exception as e:
            logger.lg.warning('unable to complete request: {}'.format(e))

    def remove_html_tags(self, data):
        """Method Name: remove_html_tags
           Description: It will remove all the html_tags present in data & returr
        try:
            pattern = re.compile('<.*?>')
            return pattern.sub(r'', data)
        except Exception as e:
            logger.lg.warning('unable to complete request: {}'.format(e))

    def remove_unwanted_bracs(self, data):
        """Method Name: remove_unwanted_bracs
           Description: It will remove all the unwanted brackets present in data
        try:
            text = re.sub(r"[\([{})\]]", "", data)
            return text
        except Exception as e:
            logger.lg.warning('unable to complete request: {}'.format(e))

    def remove_links(self, data):
        """Method Name: remove_links
           Description: It will remove all the links present in data & return cle
        try:
            text = re.sub(r'^https?:\/\/.*[\r\n]*', '', data, flags=re.MULTILINE)
            return text
        except Exception as e:
            logger.lg.warning('unable to complete request: {}'.format(e))

        return
```

```
    def remove_stop_words(self, data):
        """Method Name: remove_stop_words
           Description: It will remove all the stopwords present in data & returr
        text = data.split()
        new = []
        try:
            for i in text:
                if i not in stopwords.words('english'):
                    new.append(i)
            return " ".join(new)
        except Exception as e:
            logger.lg.warning('unable to complete request: {}'.format(e))

    def more_text_preprocessing_steps(self, data):
        """Method Name: more_text_preprocessing_steps
           Description: In future, as per the need & necessity, more text preproc
        pass
```

In [21]:
```
data=df[['Text','Star']]
run = Text_Preprocessor()
data = data.dropna(axis=0)
data["Text"] = data["Text"].apply(lambda x: run.remove_html_tags(x))
data["Text"] = data["Text"].apply(lambda x: run.remove_unwanted_bracs(x))
data["Text"] = data["Text"].apply(lambda x: run.remove_links(x))
data_text_cleaner = data["Text"].apply(lambda x: run.text_cleaner(x))
```

## *Sentimennt Analsyer*

In [22]:
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
def sentiments(data):
        analyzer = SentimentIntensityAnalyzer()
        sentiment_polarity = data.apply(lambda review: analyzer.polarity_scores(r
        compound = sentiment_polarity.apply(lambda score_dict: score_dict['compou
        return sentiment_polarity, compound
```

In [24]:
```
sentiment_polarity, compound = sentiments(data_text_cleaner)
```

In [11]:
```
data['Sentiments']  = compound.apply(lambda c: 'Positive' if c > 0.4 else ('Negat
data.head()
```

Out[11]:

| | Text | Star | Sentiments |
|---|---|---|---|
| **0** | This is very helpfull aap. | 5 | Positive |
| **1** | Good | 3 | Positive |
| **2** | Not able to update. Neither able to uninstall. | 1 | Neutral |
| **3** | Nice app | 4 | Positive |
| **4** | Many unwanted ads | 1 | Neutral |

In [12]:
```
attention_req = data[(data["Sentiments"] == "Positive") & (data["Star"] < 2)]
```

In [13]:
```
attention_req.head(10)
```

Out[13]:

| | Text | Star | Sentiments |
|---|---|---|---|
| **42** | Okk kind but bad then brave | 1 | Positive |
| **101** | Good | 1 | Positive |
| **158** | Good | 1 | Positive |
| **258** | It is the best app for browsing | 1 | Positive |
| **272** | I Depend on CHROME to GET it Right !! 💥 !! Mores... | 1 | Positive |
| **277** | Dark mode is acting up and changing screen lig... | 1 | Positive |
| **289** | Latest update turns my screen pink on some web... | 1 | Positive |
| **310** | Best | 1 | Positive |
| **312** | Good | 1 | Positive |
| **315** | Nice | 1 | Positive |

In [ ]:
```

```