```
!pip install pandas
import pandas as pd
```

```
Requirement already satisfied: pandas in /opt/python/envs/default/lib/python3.8/site-packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /opt/python/envs/default/lib/python3.8/site-packages (from pand
Requirement already satisfied: pytz>=2020.1 in /opt/python/envs/default/lib/python3.8/site-packages (from pandas) (2023.
Requirement already satisfied: numpy>=1.20.3 in /opt/python/envs/default/lib/python3.8/site-packages (from pandas) (1.24
Requirement already satisfied: six>=1.5 in /opt/python/envs/default/lib/python3.8/site-packages (from python-dateutil>=2

[notice] A new release of pip is available: 23.1.2 -> 23.3
[notice] To update, run: pip install --upgrade pip
```

```
dump=pd.read_csv("/data/notebook_files/AmazonAlexa_ReviewsDataset.tsv",sep="\t")
dump
```

	rating	date	variation	verified_reviews	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1
...	...	...	...	...	...
3145	5	30-Jul-18	Black Dot	Perfect for kids, adults and everyone in betwe...	1
3146	5	30-Jul-18	Black Dot	Listening to music, searching locations, check...	1
3147	5	30-Jul-18	Black Dot	I do love these things, i have them running my...	1
3148	5	30-Jul-18	White Dot	Only complaint I have is that the sound qualit...	1
3149	4	29-Jul-18	Black Dot	Good	1

3150 rows × 5 columns

```
dataset=dump[['verified_reviews','rating']]
dataset.columns=['Review','Sentiment']
dataset
```

	Review	Sentiment
0	Love my Echo!	5
1	Loved it!	5
2	Sometimes while playing a game, you can answer...	4
3	I have had a lot of fun with this thing. My 4 ...	5
4	Music	5
...	...	...
3145	Perfect for kids, adults and everyone in betwe...	5
3146	Listening to music, searching locations, check...	5
3147	I do love these things, i have them running my...	5
3148	Only complaint I have is that the sound qualit...	5
3149	Good	4

3150 rows × 2 columns

```
print(dataset['Sentiment'].value_counts())
```

```
5    2286
4     455
1     161
3     152
2      96
Name: Sentiment, dtype: int64
```

```
#for converting ratings to binary sentiments 0->negative and 1-> positive
def compute_sentiments(labels):
    sentiments = []
    for label in labels:
        if label > 3.0:
            sentiment = 1
        elif label <=3.0:
```

```python
        sentiment =0
    sentiments.append(sentiment)
return sentiments
```

```python
dataset['Sentiment'] = compute_sentiments(dataset.Sentiment)
dataset
```

```
<ipython-input-9-13a36a853701>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-\
  dataset['Sentiment'] = compute_sentiments(dataset.Sentiment)
```

	Review	Sentiment
0	Love my Echo!	1
1	Loved it!	1
2	Sometimes while playing a game, you can answer...	1
3	I have had a lot of fun with this thing. My 4 ...	1
4	Music	1
...	...	...
3145	Perfect for kids, adults and everyone in betwe...	1
3146	Listening to music, searching locations, check...	1
3147	I do love these things, i have them running my...	1
3148	Only complaint I have is that the sound qualit...	1
3149	Good	1

3150 rows × 2 columns

```python
print(dataset['Sentiment'].value_counts())
```

```
1    2741
0     409
Name: Sentiment, dtype: int64
```

```python
dataset.isnull().sum()
```

```
Review       0
Sentiment    0
dtype: int64
```

```python
x = dataset["Review"]
y = dataset["Sentiment"]
```

```python
#lets make a tokenizer for preprocessing with spacy
!pip install spacy
!python3 -m spacy download en_core_web_sm
```

```
Requirement already satisfied: spacy in /opt/python/envs/default/lib/python3.8/site-packages (3.7.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /opt/python/envs/default/lib/python3.8/site-packages (from
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /opt/python/envs/default/lib/python3.8/site-packages (from
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /opt/python/envs/default/lib/python3.8/site-packages (from s
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy
Requirement already satisfied: thinc<8.3.0,>=8.1.8 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /opt/python/envs/default/lib/python3.8/site-packages (from spa
Requirement already satisfied: weasel<0.4.0,>=0.1.0 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy)
Requirement already satisfied: pathy>=0.10.0 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy) (0.10.
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /opt/python/envs/default/lib/python3.8/site-packages (from sp
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /opt/python/envs/default/lib/python3.8/site-packages (from spa
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /opt/python/envs/default/lib/python3.8/site-packa
Requirement already satisfied: jinja2 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy) (3.1.2)
Requirement already satisfied: setuptools in /opt/python/envs/default/lib/python3.8/site-packages (from spacy) (68.2.2)
Requirement already satisfied: packaging>=20.0 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy) (23.
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /opt/python/envs/default/lib/python3.8/site-packages (from spa
Requirement already satisfied: numpy>=1.15.0 in /opt/python/envs/default/lib/python3.8/site-packages (from spacy) (1.24.
Requirement already satisfied: annotated-types>=0.4.0 in /opt/python/envs/default/lib/python3.8/site-packages (from pyda
Requirement already satisfied: pydantic-core==2.10.1 in /opt/python/envs/default/lib/python3.8/site-packages (from pydan
Requirement already satisfied: typing-extensions>=4.6.1 in /opt/python/envs/default/lib/python3.8/site-packages (from py
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/python/envs/default/lib/python3.8/site-packages (from re
Requirement already satisfied: idna<4,>=2.5 in /opt/python/envs/default/lib/python3.8/site-packages (from requests<3.0.0
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/python/envs/default/lib/python3.8/site-packages (from requests
```

```python
#the text_data cleaning function will be our preprocessing function for lemmatizing
import spacy
nlp = spacy.load('en_core_web_sm')
import string
punct = string.punctuation

from spacy.lang.en.stop_words import STOP_WORDS
stopwords = list(STOP_WORDS)
def text_data_cleaning(sentence):
    doc = nlp(sentence)                      # spaCy tokenize text

    tokens = [] # list of tokens
    for token in doc:
        if token.lemma_ != "-PRON-":
            temp = token.lemma_.lower().strip()
        else:
            temp = token.lower_
        tokens.append(temp)

    cleaned_tokens = []
    for token in tokens:
        if token not in stopwords and token not in punct:
            cleaned_tokens.append(token)
    return cleaned_tokens


#for using the lemma (root) of the words and convert them to lower forms
#remove punc, stopwords
#I created that function above
#I will iterate all the reviews with that function for preprocessing


exampletoken=text_data_cleaning("Those were the best days of my life")
print(exampletoken)

    ['good', 'day', 'life']


!pip install sklearn
!pip3 install scikit-learn
```

```
    Collecting sklearn
      Using cached sklearn-0.0.post10.tar.gz (3.6 kB)
      Preparing metadata (setup.py) ... done
    Building wheels for collected packages: sklearn
      Building wheel for sklearn (setup.py) ... done
      Created wheel for sklearn: filename=sklearn-0.0.post10-py3-none-any.whl size=2959 sha256=2bf959d112897b7695ee82eb76ce6
      Stored in directory: /home/datalore/.cache/pip/wheels/3b/1f/cd/e6505747b74c9bccdec4a1d8a768960cbd49d5ba8c88a58652
    Successfully built sklearn
    Installing collected packages: sklearn
    Successfully installed sklearn-0.0.post10

    [notice] A new release of pip is available: 23.1.2 -> 23.3
    [notice] To update, run: pip install --upgrade pip
    Requirement already satisfied: scikit-learn in /opt/python/envs/default/lib/python3.8/site-packages (1.2.2)
    Requirement already satisfied: numpy>=1.17.3 in /opt/python/envs/default/lib/python3.8/site-packages (from scikit-learn)
```
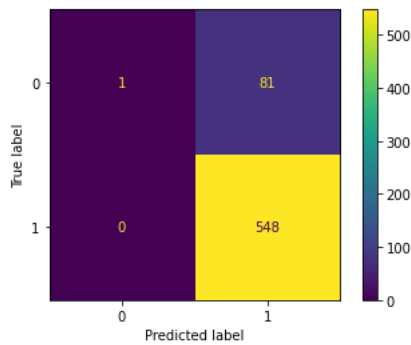
```python
#please refer to tfidf presentation
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf=TfidfVectorizer(tokenizer=text_data_cleaning)
```

```python
#lets begin with splitting our dataset to train and test sets
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42,stratify=dataset.Sentiment)
```

```python
x_train.shape, x_test.shape
```

```
((2520,), (630,))
```

```python
from sklearn.svm import LinearSVC
classifier = LinearSVC()
```

```python
#or you can use another classifier for your model creation
from sklearn.linear_model import LogisticRegression
classifier2=LogisticRegression()
```

```python
#or you can use another classifier for your model creation
from sklearn.neighbors import KNeighborsClassifier
classifier3=KNeighborsClassifier()
```

```python
from sklearn.pipeline import Pipeline
pipeline=Pipeline([('tfidf',tfidf),('clf',classifier3)])
```

```python
#it will first do vectorizer -tfidf- then classify in terms of classification method
```

```python
pipeline.fit(x_train,y_train)
```

```
/opt/python/envs/default/lib/python3.8/site-packages/sklearn/feature_extraction/text.py:528: UserWarning: The parameter '
  warnings.warn(
```

```
▸      Pipeline
  ▸ TfidfVectorizer

▸ KNeighborsClassifier
```

```python
y_pred=pipeline.predict(x_test)
print(y_pred)
```

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1]
```

```python
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, ConfusionMatrixDisplay
```

```python
cm=confusion_matrix(y_test,y_pred,labels=pipeline.classes_)
print(cm)
```

```
[[  1  81]
 [  0 548]]
```

```python
import matplotlib.pyplot as plt
disp=ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=pipeline.classes_)
disp.plot()
plt.show()
```



```python
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.01      0.02        82
           1       0.87      1.00      0.93       548

    accuracy                           0.87       630
   macro avg       0.94      0.51      0.48       630
weighted avg       0.89      0.87      0.81       630
```

```python
prediction = pipeline.predict(["Alexa is a bad tool"])
prediction
```

```
    array([1])
```

```python
import pickle
with open("model.pkl","wb") as f:
        pickle.dump(pipeline,f)
```

```python
import pickle
file_path= '/data/notebook_files/.private/model.pkl'
with open(file_path,'rb') as f:
    model=pickle.load(f)
model.predict(["alexa is good"])[0]
```

```
    ---------------------------------------------------------------------------
    Traceback (most recent call last)
        at line 3 in <module>
    FileNotFoundError: [Errno 2] No such file or directory:
    '/data/notebook_files/.private/model.pkl'
```

ARAMA YIĞINI TAŞMASI