Sentimate Model Analysis

Name : Gaikwad Pawan Ramesh

Import Necessary Libraries

```
import numpy as np
import pandas as pd
import tensorflow as tf
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
```

Read and Load the Dataset

```
DATASET_COLUMNS=['target','ids','date','flag','user','text']
DATASET_ENCODING = "ISO-8859-1"
dataset = pd.read_csv('training.1600000.processed.noemoticon.csv',encoding=DATASET_ENCODING,n
```

Exploratory Data Analysis On Dataset

Five top records of data

```
dataset.head()
```

	target	ids	date	flag	user	text
0	0	2327192646	Thu Jun 25 08:02:13 PDT 2009	NO_QUERY	quiz_master	Was having dinner with parents downstairs in D...
1	0	2327193206	Thu Jun 25 08:02:16 PDT 2009	NO_QUERY	djcampos	Blah 5am still up daang I got deep problems
2	0	2327193455	Thu Jun 25 08:02:17 PDT 2009	NO_QUERY	RKF	@jenspeedy I would suggest avoiding 360 Living

Five last records of data

```
dataset.tail()
```

	target	ids	date	flag	user	text
9995	4	1468599653	Tue Apr 07 02:39:03 PDT 2009	NO_QUERY	danalynbyers	@lbran, thanks for sending us the package - go...
9996	4	1468599688	Tue Apr 07 02:39:04 PDT 2009	NO_QUERY	joscelinyeo	@ickleoriental hahahha.. U obviously don't hv ...
			Tue Apr 07			@juliekoh It's an

Total Number of Columns in Dataset

```
dataset.columns
```

```
Index(['target', 'ids', 'date', 'flag', 'user', 'text'], dtype='object')
```

Shape of data

```
dataset.shape
```

```
(10000, 6)
```

```
dataset['target'].dtype
```

```
dtype('int64')
```

detail of dataset in max,std,count

```
dataset.describe()
```

	target	ids
count	10000.0000	1.000000e+04
mean	2.0000	1.898212e+09

Data information

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   target  10000 non-null  int64 
 1   ids     10000 non-null  int64 
 2   date    10000 non-null  object
 3   flag    10000 non-null  object
 4   user    10000 non-null  object
 5   text    10000 non-null  object
dtypes: int64(2), object(4)
memory usage: 468.9+ KB
```

```
dataset['target'].value_counts()
```

```
4    5000
0    5000
Name: target, dtype: int64
```

Checking for missing values

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

```
target    0
ids       0
date      0
flag      0
user      0
text      0
dtype: int64
```

Data Preprocessing

Selecting the text and Target column for our further analysis

```
data=dataset[['text','target']]
```

Print unique values of target

```
data['target'].unique()
```

```
array([0, 4])
```

```
dataset['user'].unique()
```

```
array(['swgalibertarian', 'captain_jen', 'UntoldElegance', ...,
       'AlexneedsJK', 'redsoxjustine', 'Celticpussycat'], dtype=object)
```

Replacing the values to ease understanding. (Assigning 1 to Positive sentiment 4)

```
dataset['target'] = dataset['target'].replace(4,1)
```

Print unique values of target

```
data['target'].unique()
```

```
array([0, 4])
```

Target class is balanced after subsetting the data

```
dataset['target'].value_counts()
```

```
0    5000
1    4999
Name: target, dtype: int64
```

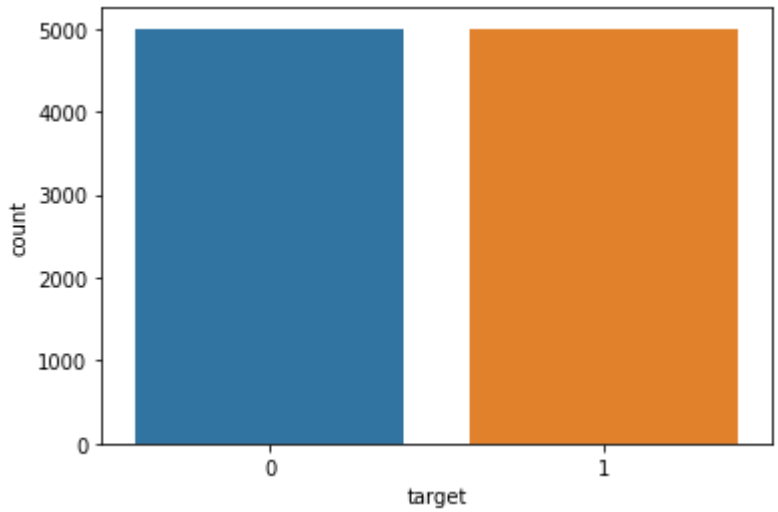
Applying the Preprocess Function on the subset data

```
dataset['text'] = dataset['text'].apply(lambda x: preprocess(x))
```

Data Visualization of Target Variables

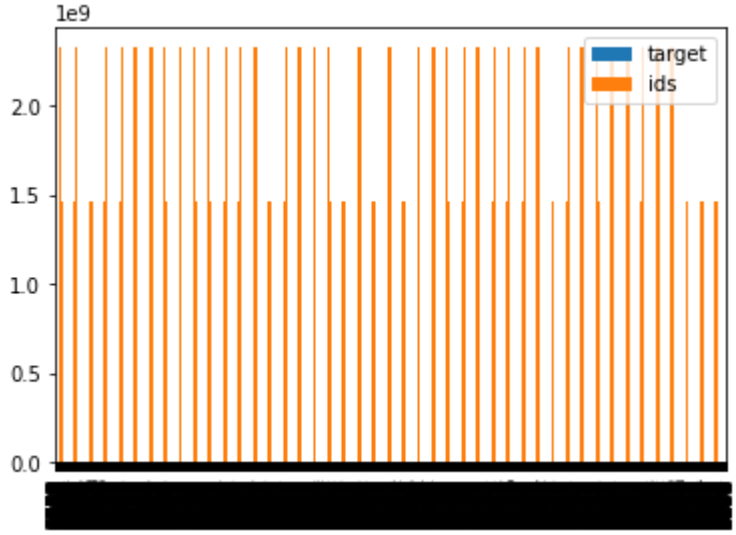
```
import seaborn as sns
sns.countplot(x='target', data=dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5099e52990>
```

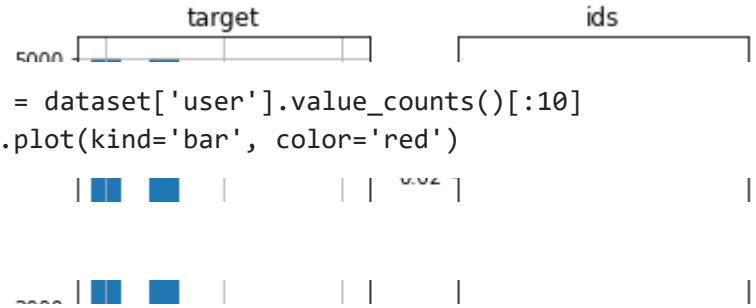



```
dataset.plot(kind = 'bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5099e61910>
```



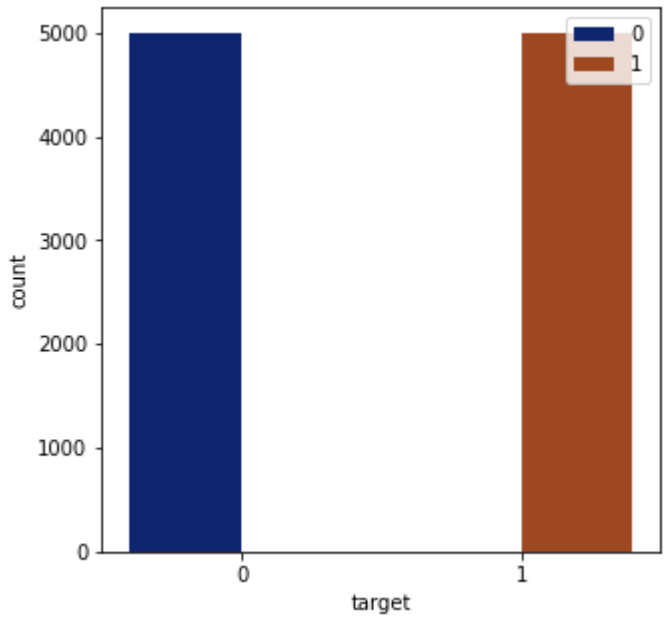

```
dataset.hist(bins=np.arange(0, 5, 0.5)-0.25)
plt.grid(False)
plt.show()
```




```
users = dataset['user'].value_counts()[:10]
users.plot(kind='bar', color='red')
```

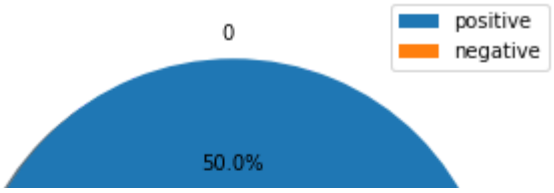


```
plt.rcParams['figure.figsize'] = (5,5)
sns.countplot(dataset["target"],hue = dataset["target"],palette = 'dark')
plt.legend(loc = 'upper right')
plt.show()
```

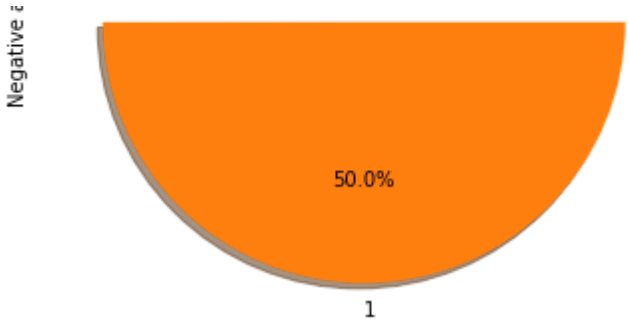



```
dataset["target"].value_counts().plot(kind = 'pie', explode = [0,0.1], figsize = (6,6), autop

plt.ylabel("Negative and Positive")
plt.legend(["positive", "negative"])
plt.show()
```



Thank you so much



✓ 0s completed at 12:15