In [ ]:

```
```

In [1]:

```
import sys
print(sys.executable)
```

```
C:\Users\teres\anaconda3\python.exe
```

In [2]:

```
#!pip install wordcloud
```

In [3]:

```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud,STOPWORDS
import nltk as nlp
import re

import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
tweets = pd.read_csv(r'C:\Users\teres\Downloads\tweets.csv')
tweets.head(3)
```

Out[4]:

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_c
0	5.700000e+17	neutral	1.0000	NaN	
1	5.700000e+17	positive	0.3486	NaN	

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_c
2	5.700000e+17	neutral	0.6837	NaN	

In [5]:
```
tweets.columns
```

Out[5]:
```
Index(['tweet_id', 'airline_sentiment', 'airline_sentiment_confidence',
       'negativereason', 'negativereason_confidence', 'airline',
       'airline_sentiment_gold', 'name', 'negativereason_gold',
       'retweet_count', 'text', 'tweet_coord', 'tweet_created',
       'tweet_location', 'user_timezone'],
      dtype='object')
```

Getting the data that we want

In [6]:
```
data = tweets[["airline_sentiment","text","airline","retweet_count"]]
data.head()
```

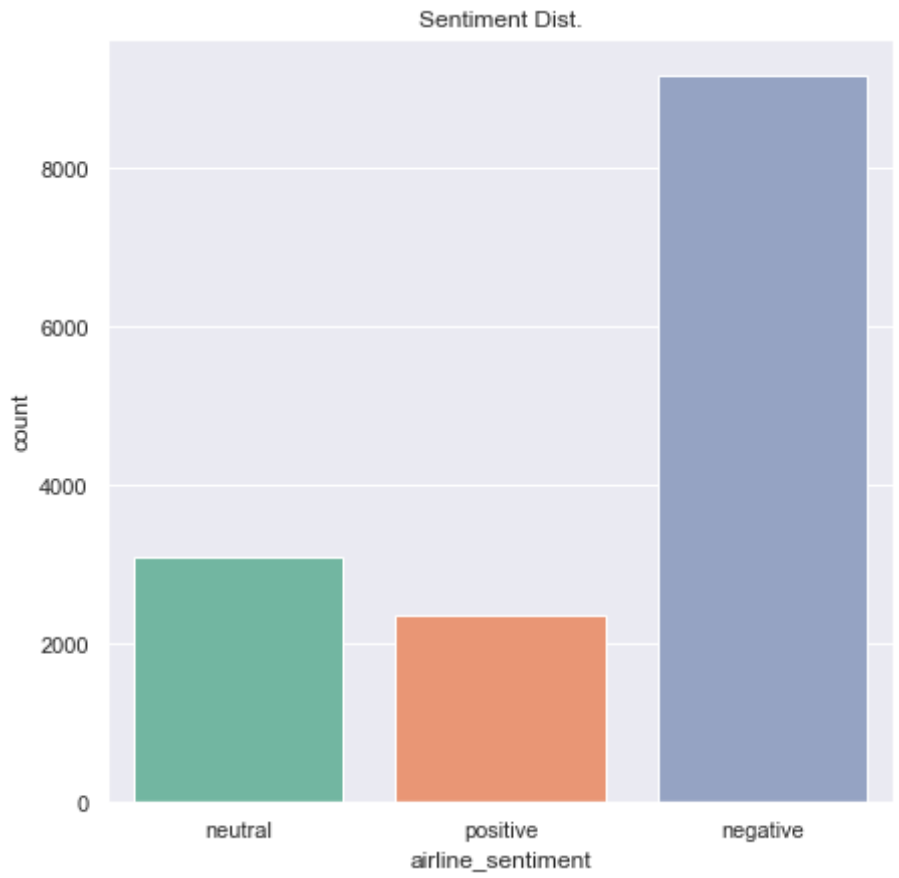
Out[6]:

	airline_sentiment	text	airline	retweet_count
0	neutral	@VirginAmerica What @dhepburn said.	Virgin America	0
1	positive	@VirginAmerica plus you've added commercials t...	Virgin America	0
2	neutral	@VirginAmerica I didn't today... Must mean I n...	Virgin America	0
3	negative	@VirginAmerica it's really aggressive to blast...	Virgin America	0
4	negative	@VirginAmerica and it's a really big bad thing...	Virgin America	0

EDA

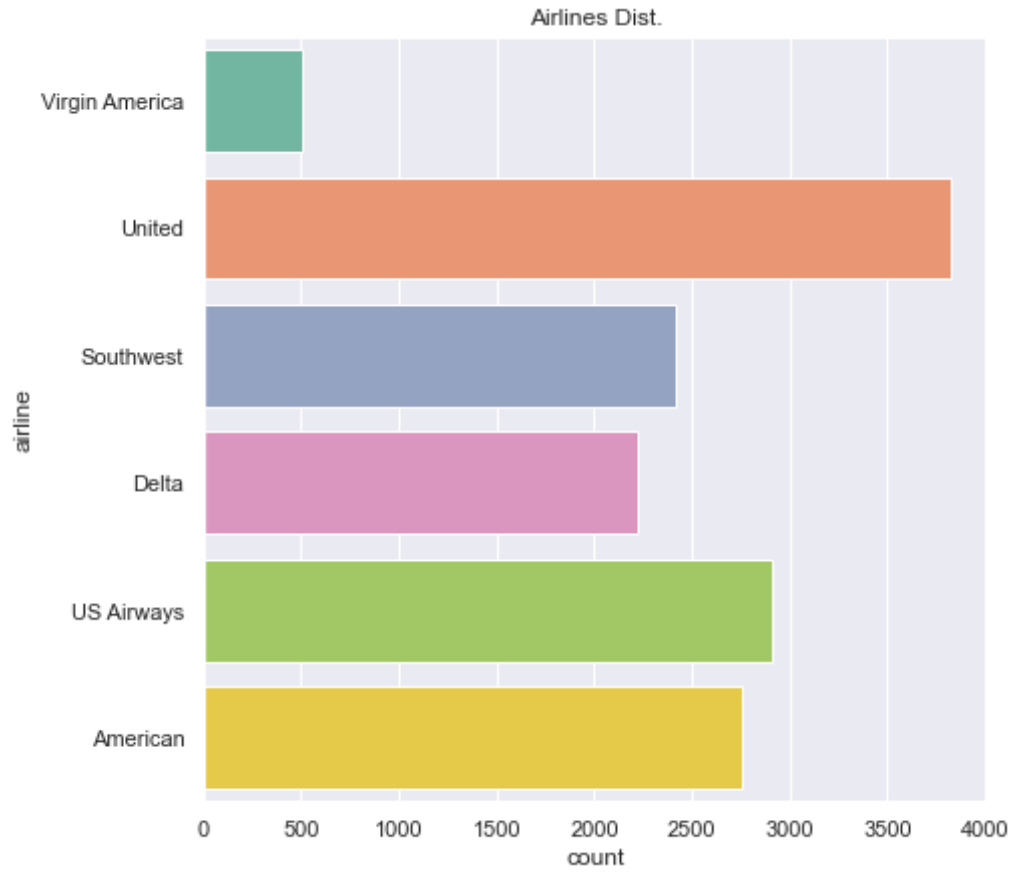
In [7]:
```
sns.set()
plt.figure(figsize=(7,7))
sns.countplot(x=data["airline_sentiment"],palette="Set2")
```

```
plt.title("Sentiment Dist.")
plt.show()
```



We see that the overall comments are negative. This may mean that you are generally unhappy with airline companies. However, this may also be due to the nature of twitter.

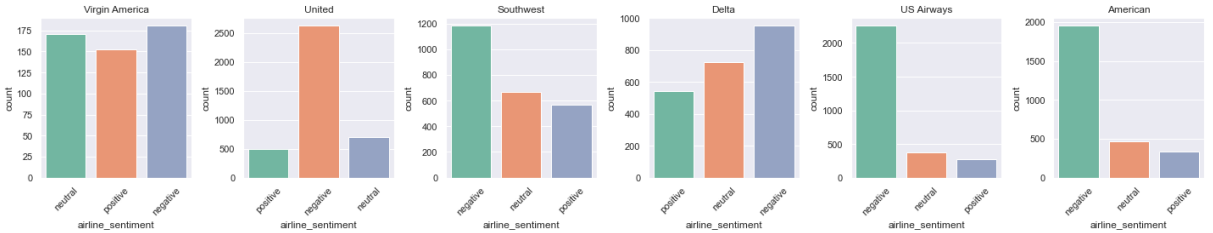
In [8]:
```
sns.set()
plt.figure(figsize=(7,7))
sns.countplot(y=data["airline"],palette="Set2")
plt.title("Airlines Dist.")
plt.show()
```



United airline is popular on Twitter. Of course, we don't know if this popularity is positive or negative. In addition, the fact that virgin america has very few twits gives the impression that the standard is neither good nor bad.

In [9]:
```
j=1
plt.subplots(figsize=(20,4),tight_layout=True)
for i in data["airline"].unique():
        x = data[data["airline"]==i]
        plt.subplot(1, 6, j)
        sns.countplot(x["airline_sentiment"],palette="Set2")
        plt.xticks(rotation=45)
```

```
        plt.title(i)
        j +=1
plt.show()
```



We are now able to comment on the emotions of the twits about companies. The United airline mentioned above has a very bad reputation in twitter. They also have negative reviews, which can be said that The United airline,US Airways and American offer poor service and have very bad reputation..

. Data cleaning and tranformation

In the following, the twits are removed from unnecessary characters, converted to lowercase letters, separated into words and their roots are obtained.

In [10]:
```
import nltk
nltk.download('punkt')
lemma = nlp.WordNetLemmatizer()
def preprocess(x):
    x = str(x)
    x = re.sub("[^a-zA-z]", " ",x)
    x = x.lower()
    x = nlp.word_tokenize(x)
    #x = [i for i in x if not i in set(stopwords.words("english"))] #sl
owly
    x = [lemma.lemmatize(i) for  i in x]
    x = " ".join(x)
    return x
```

```
data.text = data.text.apply(preprocess)
data.text[0:10]
```

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

Out[10]:

```
0                    virginamerica what dhepburn said
1    virginamerica plus you ve added commercial to ...
2    virginamerica i didn t today must mean i need ...
3    virginamerica it s really aggressive to blast ...
4    virginamerica and it s a really big bad thing ...
5    virginamerica seriously would pay a flight for...
6    virginamerica yes nearly every time i fly vx t...
7    virginamerica really missed a prime opportunit...
8           virginamerica well i didn t but now i do d
9    virginamerica it wa amazing and arrived an hou...
Name: text, dtype: object
```

We can creating word count now. We should did this in EDA but let's take a look

In [11]:

```
allcomments = " ".join(data.text)
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = STOPWORDS,
                      min_font_size = 12).generate(allcomments)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.title("All Tweets Wordcount")
plt.show()
```

At this wordcloud was talked about thats ; flights, jetblue, americanair, southwestair, usairway, customer sevices, late flights, ticket, seat, cancelled filight.

In this work, We will try predict sentiment by twits. So, get sentment and text from data.

Now:

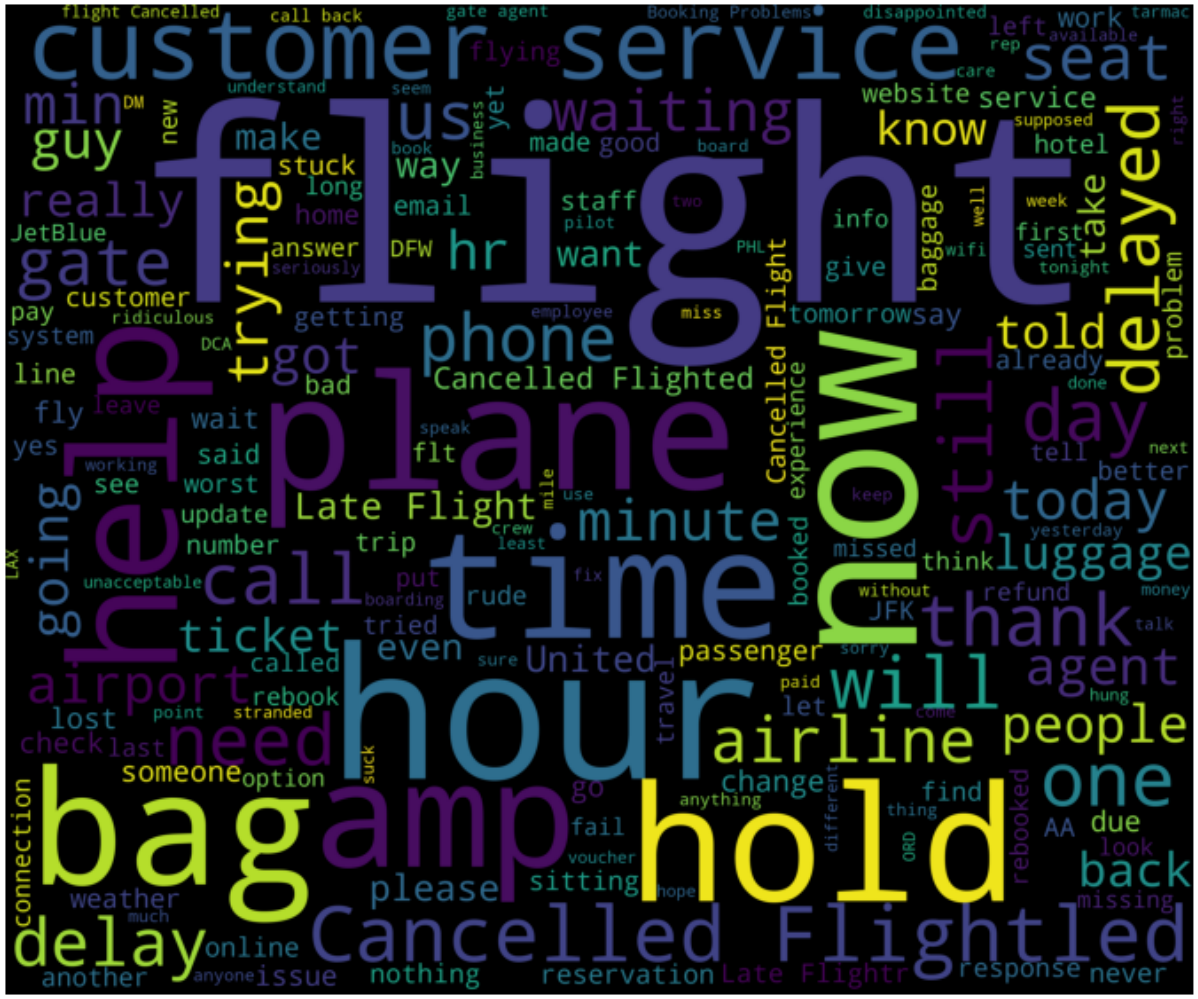
Word Cloud for the negative Tweets

In [12]:
```
from wordcloud import WordCloud,STOPWORDS
```

In [13]:
```
tweet=tweets[tweets['airline_sentiment']=='negative']
words = ' '.join(tweet['text'])
cleaned_word = " ".join([word for word in words.split()
                            if 'http' not in word
                                and not word.startswith('@')
                                and word != 'RT'
                            ])
```

In [14]:
```
wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='black',
                      width=3000,
                      height=2500
                     ).generate(cleaned_word)
```

In [15]:
```
plt.figure(1,figsize=(12, 12))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



In [16]:
```
#data=data[["airline_sentiment","text"]]
#data.head()
```

We need transform sentiments to numeric.

In [17]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

```
data["airline_sentiment"] = encoder.fit_transform(data["airline_sentiment"])
print(encoder.classes_)
data.head()
```

```
['negative' 'neutral' 'positive']
```

Out[17]:

	airline_sentiment	text	airline	retweet_count
0	1	virginamerica what dhepburn said	Virgin America	0
1	2	virginamerica plus you ve added commercial to ...	Virgin America	0
2	1	virginamerica i didn t today must mean i need ...	Virgin America	0
3	0	virginamerica it s really aggressive to blast ...	Virgin America	0
4	0	virginamerica and it s a really big bad thing ...	Virgin America	0

In [18]:

```
# convert to categorical Categority by using one hot tecnique
df_dummy = data.copy()
df_dummy.airline = pd.Categorical(df_dummy.airline)
x = df_dummy[['airline']]
del df_dummy['airline']
dummies = pd.get_dummies(x, prefix = 'airline')
data = pd.concat([df_dummy,dummies], axis=1)
data.head()
```

Out[18]:

	airline_sentiment	text	retweet_count	airline_American	airline_Delta	airline_Southwest
0	1	virginamerica what dhepburn said	0	0	0	0
1	2	virginamerica plus you ve added commercial to ...	0	0	0	0

	airline_sentiment	text	retweet_count	airline_American	airline_Delta	airline_Southwest
2	1	virginamerica i didn t today must mean i need ...	0	0	0	0
3	0	virginamerica it s really aggressive to blast ...	0	0	0	0
4	0	virginamerica and it s a really big bad thing ...	0	0	0	0

In [19]:
```
#normalize retweet count
_max = data.retweet_count.describe()[7]
data.retweet_count = [i/_max for i in data.retweet_count]
```

Model Training

Future Extraction

In [20]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words = "english")
encoded_X = vectorizer.fit_transform(data.text).toarray()
print(encoded_X.shape)
print("Features First 100:",vectorizer.get_feature_names()[:100])
```

```
(14640, 12427)
Features First 100: ['__rwg__', '_austrian', '_defcon_', '_emmaclifford', '_exact_', '_justdippin_', '_lucy_may', '_robprice', '_wtvd', 'a_life_story_', 'aa', 'aaaand', 'aaadvantage', 'aaalwayslate', 'aacustomerservice', 'aadavantage', 'aadelay', 'aadv', 'aadvantage', 'aafail', 'aakjumxa', 'aal', 'aaron', 'aarp', 'aateam', 'aau', 'aavvoreph', 'aay', 'a
```

```
b', 'aback', 'abandon', 'abandoned', 'abandonment', 'abassinet', 'abbre
ve', 'abc', 'abcletjetbluestreamfeed', 'abcnetwork', 'abcnews', 'abduct
ed', 'abi', 'abigailedge', 'ability', 'able', 'aboard', 'aboout', 'abou
nds', 'abprg', 'abq', 'abroad', 'absolute', 'absolutely', 'absorb', 'ab
sorber', 'absoulutely', 'absurd', 'absurdity', 'absurdly', 'abt', 'abtw
f', 'abundance', 'abuse', 'abused', 'abxrq', 'abysmal', 'ac', 'acarl',
'acc', 'accelerate', 'accept', 'acceptable', 'accepted', 'accepting',
'acces', 'access', 'accessibility', 'accessible', 'accessing', 'acciden
t', 'accidentally', 'accomidating', 'accommodate', 'accommodated', 'acc
ommodates', 'accommodating', 'accommodation', 'accompaniment', 'accompa
ny', 'accomplish', 'accomplished', 'according', 'accordingly', 'accoun
t', 'accountability', 'accountable', 'accrue', 'accruing', 'acct', 'acc
ts', 'accumulation']
```

In [21]:
```
data2 = data.copy()
del data2["text"]
data2 = pd.concat([pd.DataFrame(encoded_X),data2], axis=1)
data2.head()
```

Out[21]:

	0	1	2	3	4	5	6	7	8	9	...	12425	12426	airline_sentiment	retweet_cou
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1	0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	2	0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1	0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0	0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0	0

5 rows × 12435 columns

In [22]:
```
X = data2.drop(["airline_sentiment"],axis=1)
y = data2.airline_sentiment
```

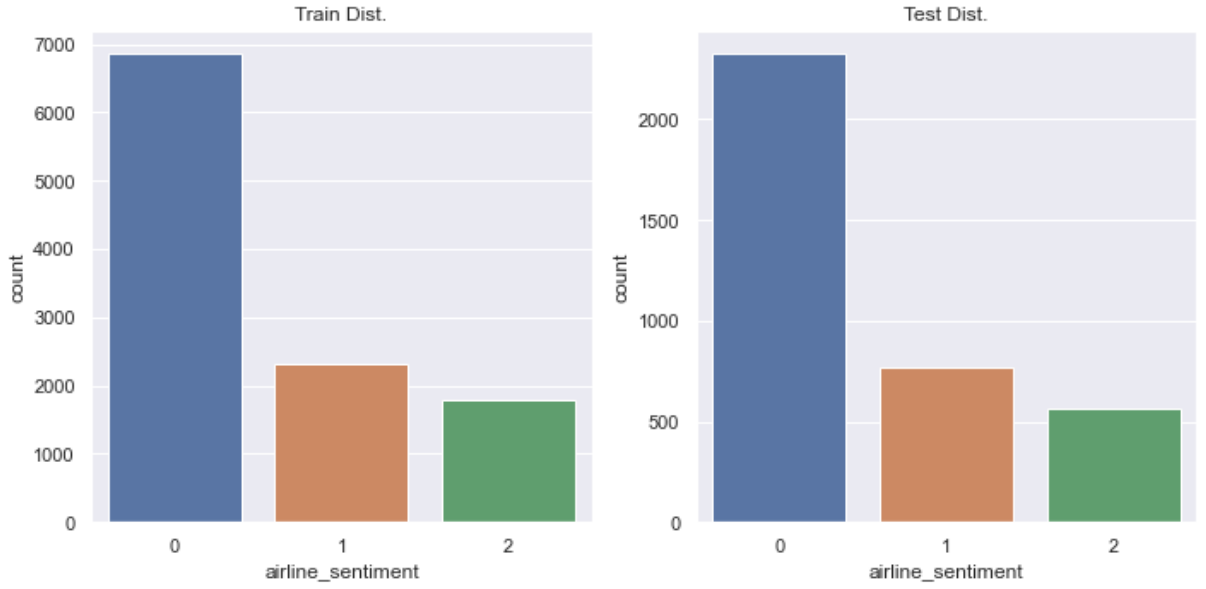
Split train and test data

In [23]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.25,
random_state=22)
print("Train :",X_train.shape)
print("Test  :",X_test.shape)
```

```
Train : (10980, 12434)
Test  : (3660, 12434)
```

Check test distribution

In [24]:
```
sns.set()
plt.subplots(figsize=(10,5),tight_layout=True)
plt.subplot(1,2,1)
sns.countplot(y_train)
plt.title("Train Dist.")
plt.subplot(1,2,2)
sns.countplot(y_test)
plt.title("Test Dist.")
plt.show()
```



Random Forest classifier

In [27]:
```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=100)
clf.fit(X_train, y_train)
pred = clf.predict(X_test)
```

Model Evaluation

In [26]:
```
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score

rf_acc = accuracy_score(y_test, pred)
rf_f1 = f1_score(y_test, pred, average="micro")

print("Random Forest")
print("Accuracy : %",round(rf_acc*100,2))
print("F1 Score : %",round(rf_f1*100,2))
```

```
Random Forest
Accuracy : % 76.23
F1 Score : % 76.23
```

In [ ]:
```

```