```
import os
import pandas as pd
from google.colab import files
uploaded = files.upload()
import pandas as pd
import io
df = pd.read_csv(io.StringIO(uploaded['news.csv'].decode('utf-8'))).head(5)
df
```

Choose Files news.csv

- **news.csv**(text/csv) - 30696129 bytes, last modified: 9/13/2019 - 100% done

```
Saving news.csv to news.csv
```

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL
5	6903	Tehran, USA	\nI'm not an immigrant, but my grandparents ...	FAKE
		Girl Horrified At What She	Share This Baylee Luciani (left)	

Next steps: Generate code with df View recommended plots

```
!pip install lime
!pip install plotly
import re
re.compile('<title>(.*)</title>')
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
import time
from sklearn.metrics import classification_report
from sklearn.compose import ColumnTransformer
from nltk.corpus import stopwords
import string
import nltk
from os import path
import numpy as np
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
from nltk.util import ngrams
nltk.download('punkt')
nltk.download('stopwords')
import lime
import sklearn.ensemble
from __future__ import print_function
from lime import lime_text
from sklearn.pipeline import make_pipeline
from lime.lime_text import LimeTextExplainer
import plotly.graph_objs as go
import matplotlib.pyplot as plt
from plotly.subplots import make_subplots
import plotly.offline as pyo
pyo.init_notebook_mode()
from plotly.offline import iplot,init_notebook_mode
init_notebook_mode()
from textblob import TextBlob
from sklearn.metrics import roc_curve, auc
```

```
Requirement already satisfied: lime in /usr/local/lib/python3.10/dist-packages (0.2.0
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages 
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from 
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: scikit-image>=0.12 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,!=8.3.0,>=6.1.0 in /usr/local/li
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (f
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (5.1
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
handles=r'@\w+'
generic_urls=r'http[s]?://(?:[A-Za-z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9A-Fa-f][0-9A-Fa-f

'''
Takes in text data and performs the following:
lowers the case of text
removes URLs
Removes non alphanumeric text and twitter handles
Tokenizes the text
'''
def preprocess(df_text):
    df_text=df_text.lower()
    df_text=df_text.replace("'","")
    df_text=df_text.replace("“","")
    df_text=df_text.replace("”","")
    df_text=df_text.replace('—','')
    df_text=df_text.replace('’','')
    df_text=df_text.replace('‘','')
    df_text= re.sub(generic_urls, '', df_text) # remove URLs
    re.sub(r'\W+', '', df_text)
    df_text= re.sub(handles, '', df_text) #remove Twitter handles
    df_text = re.sub('<[^>]*>', '', df_text)# remove none alphanumeric text
    df_text = re.sub('<.*?>+', '', df_text)
    df_text = re.sub('[%s]' % re.escape(string.punctuation), '', df_text)
    tokenized_list = nltk.word_tokenize(df_text)
    stop_words=set(stopwords.words('english'))
    cleaned_list=[i for i in tokenized_list if i not in stop_words]# remove stop words
    return ' '.join(cleaned_list)
df['cleaned_text'] = df['text'].apply(preprocess)
df['cleaned_text'].head()
```

```
0    daniel greenfield shillman journalism fellow f...
1    google pinterest digg linkedin reddit stumbleu...
2    us secretary state john f kerry said monday st...
3    kaydee king november 9 2016 lesson tonights de...
4    primary day new york frontrunners hillary clin...
Name: cleaned_text, dtype: object
```

```
'''
This function takes in a text data and returns the polarity of the text
Polarity is float which lies in the range of [-1,1] where 1 means positive statement
and -1 means a negative statement
'''
def polarity_score(df_text):
  return TextBlob(df_text).sentiment.polarity

'''
This function takes in a text data and returns the subectivity of the text.
Subjectivity sentences generally refer to personal opinion,
emotion or judgment whereas objective refers to factual information.
Subjectivity is also a float which lies in the range of [0,1].
'''
def subjectivity_score(df_text):
  return TextBlob(df_text).sentiment.subjectivity

#apply above functions to the data
df['polarity_score']=df['cleaned_text'].apply(polarity_score)
df['subjectivity_score']=df['cleaned_text'].apply(subjectivity_score)
```

```
df['polarity_score'].mean()
df['subjectivity_score'].mean()
print(' The overall polarity of the tweet data is '+
      str(round(df['polarity_score'].mean(),2)))
print(' The overall subjectivity of the tweet data is '+
      str(round(df['subjectivity_score'].mean(),2)))
```

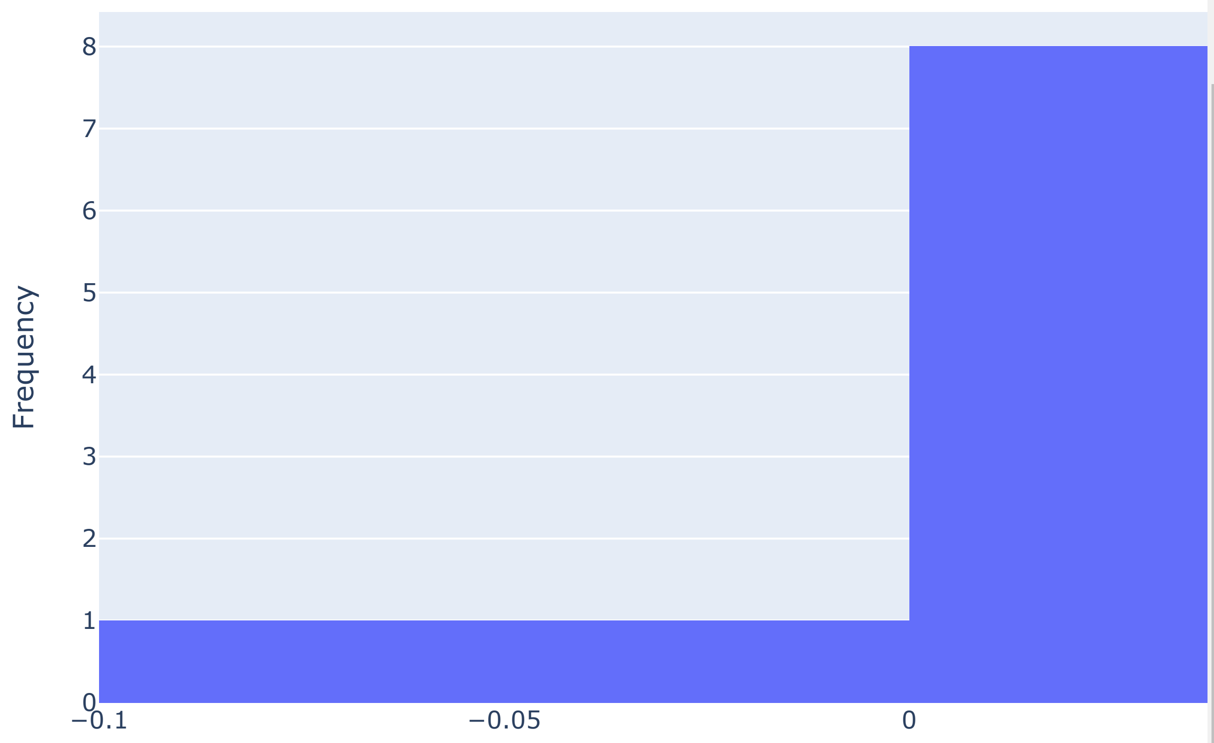
```
 The overall polarity of the tweet data is 0.07
 The overall subjectivity of the tweet data is 0.42
```

```
def configure_plotly_browser_state():
  import IPython
  display(IPython.core.display.HTML('''
        <script src="/static/components/requirejs/require.js"></script>
        <script>
          requirejs.config({
            paths: {
              base: '/static/base',
              plotly: 'https://cdn.plot.ly/plotly-latest.min.js?noext',
            },
          });
        </script>
        '''))
```

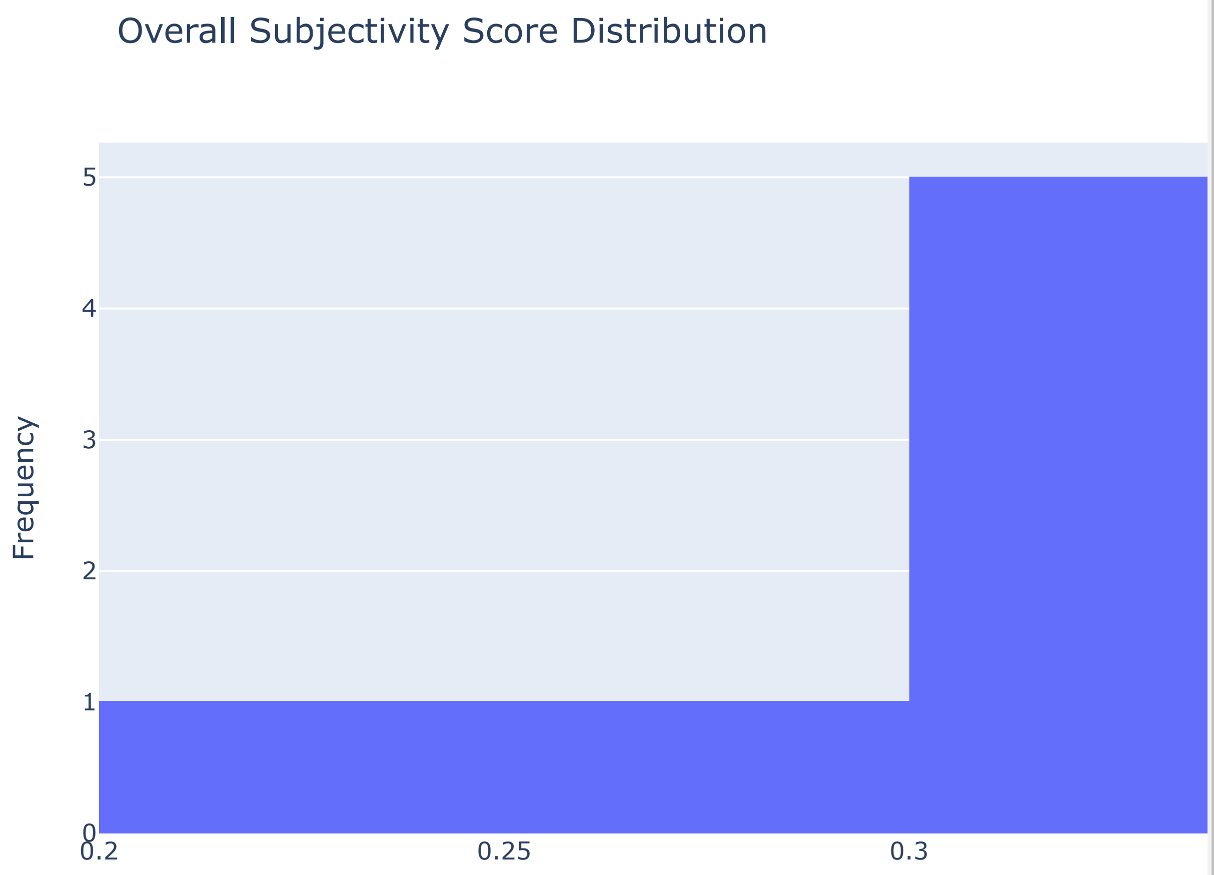
```
#Histogram Plot For Overall Polarity Distribution#

configure_plotly_browser_state()
init_notebook_mode(connected=False)
trace=go.Histogram(x=df.polarity_score)
data=trace
layout={'title':'Overall Polaritry Score Distribution',
        'xaxis':{'title':'Polarity Score'},'yaxis':{'title':'Frequency'}}
iplot({'data':data,'layout':layout})

#Histogram Plot For Overall Subjectivity Distribution#
configure_plotly_browser_state()
init_notebook_mode(connected=False)
trace=go.Histogram(x=df.subjectivity_score)
data=trace
layout={'title':'Overall Subjectivity Score Distribution',
        'xaxis':{'title':'Subjectivity Score'},'yaxis':{'title':'Frequency'}}
iplot({'data':data,'layout':layout})
```



Frequency
8
7
6
5
4
3
2
1
0
−0.1
−0.05
0



Overall Subjectivity Score Distribution
Frequency
5
4
3
2
1
0
0.2
0.25
0.3

```
pol=df[['label','polarity_score','subjectivity_score']]
pol.groupby('label').mean()
```

label	polarity_score	subjectivity_score
FAKE	0.071263	0.472312
REAL	0.076951	0.360349

```
fig = make_subplots(rows=2, cols=2,
                    subplot_titles=("Polarity Score Distribution-REAL", "Subjectivity Scc
                    x_title="Score",y_title='Frequency')
fig.add_trace(
    go.Histogram(x=df[df['label']=='REAL']['polarity_score']),
    row=1, col=1)#name="Polarity-REAL"
fig.add_trace(
    go.Histogram(x=df[df['label']=='REAL']['subjectivity_score']),
    row=1, col=2)#
fig.add_trace(
    go.Histogram(x=df[df['label']=='FAKE']['polarity_score']),
    row=2, col=1)#name="Subjectivity-REAL",
fig.add_trace(
    go.Histogram(x=df[df['label']=='FAKE']['subjectivity_score']),
    row=2, col=2)
```

```
# Tweet outliers
print('Tweet with the higest polarity:',(df[df['polarity_score']==df['polarity_score'].ma

print('Tweet with the lowest polarity:',(df[df['polarity_score']==df['polarity_score'].mi

print('Tweet with the higest subjectivity:',(df[df['subjectivity_score']==df['subjectivit

print('Tweet with the lowest subjectivity:',(df[df['subjectivity_score']==df['subjectivit
```

```
Tweet with the higest polarity:
It's primary day in New York and front-runners Hillary Clinton and Donald Trump are

Trump is now vowing to win enough delegates to clinch the Republican nomination anc

A big win in New York could tip the scales for both the Republican and Democratic f

"We have won eight out of the last nine caucuses and primaries! Cheer!" Sanders rec

While wins in New York for Trump and Clinton are expected, the margins of those vic

Trump needs to capture more than 50 percent of the vote statewide if he wants to be

"We've got to vote and you know Cruz is way, way down in the polls," Trump urged su

Meanwhile, Sanders is hoping for a close race in the Empire State. A loss by 10 poi
```

```
Despite a predicted loss in New York, Cruz hasn't lost momentum. He's hoping to swe

"Because if I'm the nominee, we win the General Election," Cruz promised his suppor

For now, Cruz, Kasich, and Sanders have all moved on from New York to other states.
Label of tweet with highest polarity:
REAL
Tweet with the lowest polarity:
A Czech stockbroker who saved more than 650 Jewish children from Nazi Germany has c
Label of tweet with lowest polarity:
REAL
Tweet with the higest subjectivity:
Daniel Greenfield, a Shillman Journalism Fellow at the Freedom Center, is a New Yor
In the final stretch of the election, Hillary Rodham Clinton has gone to war with t
The word “unprecedented” has been thrown around so often this election that it ough
But that’s exactly what Hillary and her people have done. Coma patients just waking
The FBI is under attack by everyone from Obama to CNN. Hillary’s people have circul
The FBI’s leadership is being warned that the entire left-wing establishment will f
The covert struggle between FBI agents and Obama’s DOJ people has gone explosively
The New York Times has compared Comey to J. Edgar Hoover. Its bizarre headline, “Ja
James Carville appeared on MSNBC to remind everyone that he was still alive and ins
Countless media stories charge Comey with violating procedure. Do you know what’s a
Senator Harry Reid has sent Comey a letter accusing him of violating the Hatch Act.
If James Comey is really out to hurt Hillary, he picked one hell of a strange way t
Not too long ago Democrats were breathing a sigh of relief when he gave Hillary Cli
Either Comey is the most cunning FBI director that ever lived or he’s just awkwardl
The only truly mysterious thing is why Hillary and her associates decided to go to
And it’s an interesting question.
Hillary’s old strategy was to lie and deny that the FBI even had a criminal investi
Pretending that nothing was wrong was a bad strategy, but it was a better one that
There are two possible explanations.
Hillary Clinton might be arrogant enough to lash out at the FBI now that she believ
But the other explanation is that her people panicked.
Going to war with the FBI is not the behavior of a smart and focused presidential c
During the original FBI investigation, Hillary Clinton was confident that she could
There’s only one reason for such bizarre behavior.
The Clinton campaign has decided that an FBI investigation of the latest batch of e
Clinton loyalists rigged the old investigation. They knew the outcome ahead of time
You can smell the fear.
```

```python
def tokenize_text(df_text):
  return df_text.split()
df['tokenized_tweet']=df.cleaned_text.apply(tokenize_text)
df.head()
```

	Unnamed: 0	title	text	label	cleaned_text	polarity_score	subjectiv
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	daniel greenfield shillman journalism fellow f...	0.046885	
1	10294	Watch The Exact Moment Paul Ryan Committed	Google Pinterest Digg Linkedin Reddit Stumbleu	FAKE	google pinterest digg linkedin reddit stumbleu	0.099383	

Next steps: Generate code with df View recommended plots

```
 #Word Frequency Dataframe###
word_freq=pd.DataFrame(df['cleaned_text'].str.split(expand=True).stack().value_counts()).
word_freq=word_freq.rename(columns={'index':'Word', 0:'Count'})

## Bar graph- Word Frequency For Overall Tweet Data##
configure_plotly_browser_state()
init_notebook_mode(connected=False)
trace=go.Bar(x=word_freq['Word'][0:20],y=word_freq['Count'][0:20])
data=trace
layout={'title':'Top 20 most Frequent words in across entire tweet data', 'xaxis':{'title
iplot({'data':data,'layout':layout})

     ## Bar graphs - Word Frequency Per Label ##
## Fake News Word Frequency
word_freq_Fake=pd.DataFrame(df[df['label']=='FAKE']['cleaned_text'].str.split(expand=True
word_freq_Fake=word_freq_Fake.rename(columns={'index':'Word',0:'Count'})

## Real News Word frequency
word_freq_Real=pd.DataFrame(df[df['label']=='REAL']['cleaned_text'].str.split(expand=True
word_freq_Real=word_freq_Real.rename(columns={'index':'Word',0:'Count'})

configure_plotly_browser_state()
fig = make_subplots(rows=1, cols=2,
                    subplot_titles=("Top 20 most frequent words-Fake", "Top 20 most frequ
                    x_title="Word",y_title='Frequency')
fig.add_trace(
    go.Bar(x=word_freq_Fake['Word'].iloc[0:20], y=word_freq_Fake['Count'].iloc[0:20]),
    row=1, col=1)
fig.add_trace(
    go.Bar(x=word_freq_Real['Word'].iloc[0:20], y=word_freq_Real['Count'].iloc[0:20]),
    row=1, col=2)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self,
key)
   3652         try:
-> 3653             return self._engine.get_loc(casted_key)
   3654         except KeyError as err:

                          4 frames

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Count'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self,
key)
   3653             return self._engine.get_loc(casted_key)
```

Next steps: Explain error

```
Start coding or generate with AI.
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self,
key)
   3652         try:
-> 3653             return self._engine.get_loc(casted_key)
   3654         except KeyError as err:

                          4 frames

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in
pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'Count'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self,
key)
   3653             return self._engine.get_loc(casted_key)
   3654         except KeyError as err:
-> 3655             raise KeyError(key) from err
```