In [1]:

```python
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import OneHotEncoder, MinMaxScaler
from sklearn.compose import make_column_transformer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
```

In [2]:

```python
tweet = pd.read_csv("Tweets.csv")
len(tweet)
```

Out[2]:

```
14640
```

In [3]:

```python
tweet.head()
```

Out[3]:

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	nega
0	570306133677760513	neutral	1.0000	NaN	
1	570301130888122368	positive	0.3486	NaN	
2	570301083672813571	neutral	0.6837	NaN	
3	570301031407624196	negative	1.0000	Bad Flight	
4	570300817074462722	negative	1.0000	Can't Tell	

In [4]:

```
tweet.describe()
```

Out[4]:

	tweet_id	airline_sentiment_confidence	negativereason_confidence	retweet_count
count	1.464000e+04	14640.000000	10522.000000	14640.000000
mean	5.692184e+17	0.900169	0.638298	0.082650
std	7.791112e+14	0.162830	0.330440	0.745778
min	5.675883e+17	0.335000	0.000000	0.000000
25%	5.685592e+17	0.692300	0.360600	0.000000
50%	5.694779e+17	1.000000	0.670600	0.000000
75%	5.698905e+17	1.000000	1.000000	0.000000
max	5.703106e+17	1.000000	1.000000	44.000000

In [5]:

```
tweet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14640 entries, 0 to 14639
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   tweet_id                      14640 non-null  int64  
 1   airline_sentiment             14640 non-null  object 
 2   airline_sentiment_confidence  14640 non-null  float64
 3   negativereason                9178 non-null   object 
 4   negativereason_confidence     10522 non-null  float64
 5   airline                       14640 non-null  object 
 6   airline_sentiment_gold        40 non-null     object 
 7   name                          14640 non-null  object 
 8   negativereason_gold           32 non-null     object 
 9   retweet_count                 14640 non-null  int64  
 10  text                          14640 non-null  object 
 11  tweet_coord                   1019 non-null   object 
 12  tweet_created                 14640 non-null  object 
 13  tweet_location                9907 non-null   object 
 14  user_timezone                 9820 non-null   object 
dtypes: float64(2), int64(2), object(11)
memory usage: 1.7+ MB
```

In [7]:

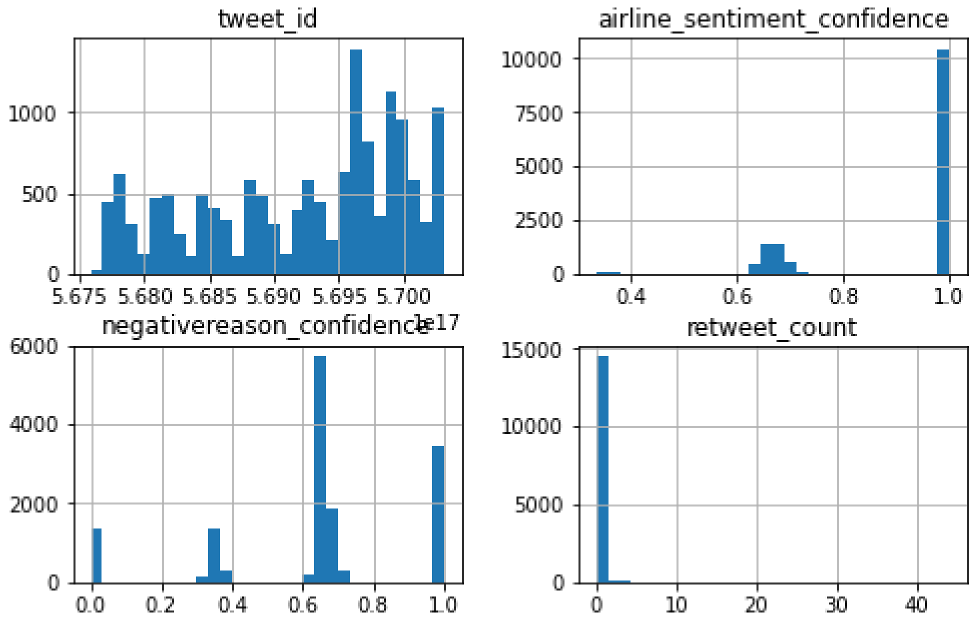
```
def deal_missing_values(X_full):
    #drop col where data is very less
    X_full = X_full.drop('airline_sentiment_gold', axis=1)
    X_full = X_full.drop('negativereason_gold', axis=1)
    X_full = X_full.drop('tweet_coord', axis=1)
    # replace null values with mean
    X_full['negativereason_confidence'] = X_full['negativereason_confidence'].fillna(X_f
    return X_full

tweet = deal_missing_values(tweet)
tweet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14640 entries, 0 to 14639
Data columns (total 12 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   tweet_id                      14640 non-null  int64  
 1   airline_sentiment             14640 non-null  object 
 2   airline_sentiment_confidence  14640 non-null  float64
 3   negativereason                9178 non-null   object 
 4   negativereason_confidence     14640 non-null  float64
 5   airline                       14640 non-null  object 
 6   name                          14640 non-null  object 
 7   retweet_count                 14640 non-null  int64  
 8   text                          14640 non-null  object 
 9   tweet_created                 14640 non-null  object 
 10  tweet_location                9907 non-null   object 
 11  user_timezone                 9820 non-null   object 
dtypes: float64(2), int64(2), object(8)
memory usage: 1.3+ MB
```

In [8]:

```
tweet.hist(bins = 30, figsize = (8,5))
plt.show()
```



In [9]:

```
(tweet['airline'].unique())
```

Out[9]:

```
array(['Virgin America', 'United', 'Southwest', 'Delta', 'US Airways',
       'American'], dtype=object)
```

In [10]:

```
(tweet['negativereason'].unique())
```

Out[10]:

```
array([nan, 'Bad Flight', "Can't Tell", 'Late Flight',
       'Customer Service Issue', 'Flight Booking Problems',
       'Lost Luggage', 'Flight Attendant Complaints', 'Cancelled Flight',
       'Damaged Luggage', 'longlines'], dtype=object)
```

In [11]:

```
tweet.tail()
```

Out[11]:

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason
14635	569587686496825344	positive	0.3487	NaN
14636	569587371693355008	negative	1.0000	Customer Service Issue
14637	569587242672398336	neutral	1.0000	NaN
14638	569587188687634433	negative	1.0000	Customer Service Issue
14639	569587140490866689	neutral	0.6771	NaN

In [19]:

```
X = tweet.drop('airline_sentiment', axis = 1)
y = tweet['airline_sentiment']
```

In [20]:

```
from sklearn.compose import make_column_transformer
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder
from sklearn.model_selection import train_test_split

ct = make_column_transformer(
    (MinMaxScaler(), ["tweet_id"]),
    (OneHotEncoder(handle_unknown="ignore"), ["airline", "retweet_count"])
)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42

ct.fit(X_train)
X_train_normal = ct.transform(X_train)
X_test_normal = ct.transform(X_test)
```

In [22]:

```
lr_model = LogisticRegression(max_iter = 1000)
lr_model.fit(X_train_normal, y_train)
tree_model = SVC()
tree_model.fit(X_train_normal, y_train)
```

Out[22]:

```
▾ SVC

SVC()
```

In [23]:

```
y_pred = lr_model.predict(X_test_normal)
accuracy = accuracy_score(y_test, y_pred)
y_pred_tree = tree_model.predict(X_test_normal)
accuracy_tree = accuracy_score(y_test, y_pred_tree)
print("Accuracy: ", accuracy_tree)
```

```
Accuracy:  0.6454918032786885
```