In [72]:
```
#visualizing google trends data

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [4]:
```
google_trends = pd.read_csv("C:\\Users\\saiei\\Downloads\\multiTimeline clean

google_trends
```

Out[4]:

| | Month | Search Engine Optimization | Freelance |
|---|---|---|---|
| **0** | 2012-12 | 27 | 15 |
| **1** | 2013-01 | 32 | 21 |
| **2** | 2013-02 | 33 | 19 |
| **3** | 2013-03 | 32 | 17 |
| **4** | 2013-04 | 33 | 18 |
| **...** | ... | ... | ... |
| **115** | 2022-07 | 86 | 19 |
| **116** | 2022-08 | 93 | 20 |
| **117** | 2022-09 | 87 | 19 |
| **118** | 2022-10 | 78 | 17 |
| **119** | 2022-11 | 80 | 18 |

120 rows × 3 columns

In [13]:
```
#cleaning the columns

google_trends=google_trends.rename(columns={'Search Engine Optimization ':' S
```

In [11]:
```
google_trends.columns=[s.strip().replace(' ','_') for s in google_trends.colu
```

In [14]:
```
google_trends.columns
```

Out[14]:
```
Index(['Month', 'Search_Engine_Optimization', 'Freelance'], dtype='object')
```

In [75]:
```
#converting month column to date formate

google_trends['Month'] = pd.to_datetime(google_trends['Month'], errors='coerc
```

In [80]:
```
google_trends.dtypes
```

Out[80]:
```
Month                          datetime64[ns]
Search_Engine_Optimization              int64
Freelance                               int64
dtype: object
```

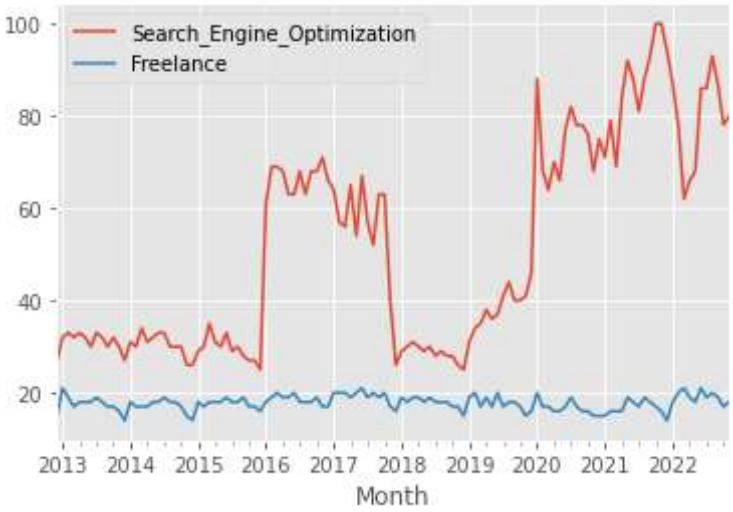
In [88]:
```
def plot_data(df):

    fig = plt.figure(figsize = (15,8))
    ax = fig.add_subplot(111)
    df.plot(ax=ax)
    plt.ylabel('Search_Engine_Optimization')
    plt.ylabel('Freelance')
    plt.xlabel('Month')
    plt.ylim((0,120))
    plt.legend(loc='lower left')
    return ax
```

In [90]:
```
#plotting google trends (USA)

plt.style.use('ggplot')
google_trends.plot.line(x='Month')
```

Out[90]:
```
<AxesSubplot:xlabel='Month'>
```




In [95]:
```
#Google trends (world)

world_trends = pd.read_csv("C:\\Users\\saiei\\Downloads\\multiTimeline_World.
```

In [96]:
```
world_trends.head()
```

Out[96]:

| | Month | Search Engine Optimization | Freelance |
|---|---|---|---|
| **0** | 2012-12 | 40 | 15 |
| **1** | 2013-01 | 48 | 19 |
| **2** | 2013-02 | 47 | 19 |
| **3** | 2013-03 | 47 | 17 |
| **4** | 2013-04 | 48 | 18 |

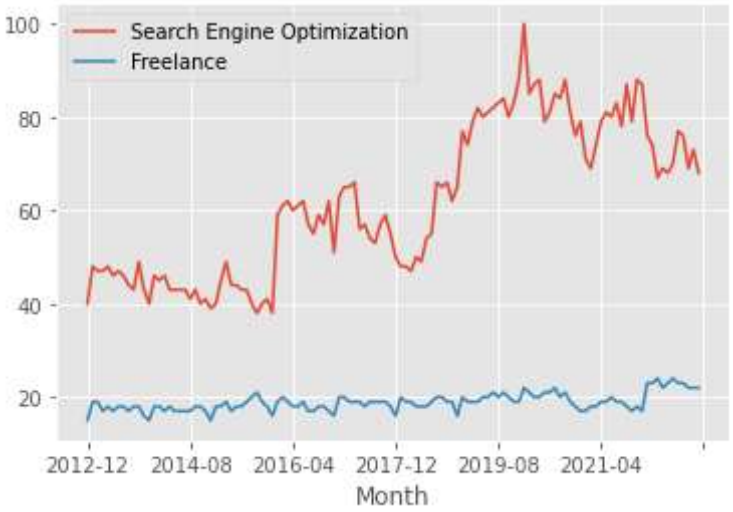
In [97]:
```
def plot_data(df):

    fig = plt.figure(figsize = (15,8))
    ax = fig.add_subplot(111)
    df.plot(ax=ax)
    plt.ylabel('Search Engine Optimization')
    plt.ylabel('Freelance')
    plt.xlabel('Month')
    plt.ylim((0,120))
    plt.legend(loc='lower left')
    return ax
```

In [98]:
```
#plotting google trends (world)

plt.style.use('ggplot')
world_trends.plot.line(x='Month')
```

Out[98]: <AxesSubplot:xlabel='Month'>




In [ ]:
```
#Visualizing Twitter data
```

In [38]:
```
import pandas as pd
```

In [39]:
```
tweets = pd.read_csv("C:\\Users\\saiei\\Downloads\\Twitter_data.csv")

tweets.head()
```

Out[39]:

| | Tweet Id | Text | Name | Screen Name | UTC |
|---|---|---|---|---|---|
| 0 | 1597027289266475008 | SEO 2021 Learn Search Engine Optimization With... | Hadley_EichmannPic | eichmannpic | 2022-11-28T00:39:11.000Z |
| 1 | 1597026923854454785 | What is Search Engine Optimization, and how ma... | ABDUL KHALAKE IBNA AHMMED | sohelahmmeddm | 2022-11-28T00:37:44.000Z |
| 2 | 1597025184619249664 | RT @CdkLabs : Investing in search engine optim... | Hello ✌️❤️ | CasualViewer123 | 2022-11-28T00:30:49.000Z |
| 3 | 1597020230886330369 | With over twenty years of experience designing... | Search Engine Social Media Agency Bristol | capidhouser | 2022-11-28T00:11:08.000Z |
| 4 | 1597019198755983360 | Looking for a Jr. Search Engine Optimization S... | Tampa SEO Academy | TampaSEOAcademy | 2022-11-28T00:07:02.000Z |

In [59]:
```
#cleaning the columns

tweets=tweets.rename(columns={'Tweet Id ':' Tweet_id'})

tweets.columns=[s.strip().replace(' ','_') for s in tweets.columns]

tweets.columns
```

Out[59]:
```
Index(['Tweet_Id', 'Text', 'Name', 'Screen_Name', 'UTC', 'Created_At',
       'Favorites', 'Retweets', 'Language', 'Client', 'Tweet_Type', 'URLs',
       'Hashtags', 'Mentions', 'Media_Type', 'Media_URLs', 'Unnamed:_16',
       'Unnamed:_17', 'Unnamed:_18'],
      dtype='object')
```

In [69]:
```
#cleaning the date

tweets['Date_1']= pd.to_datetime(tweets['Created_At'], errors='coerce')
```

In [70]:
```
tweets['Date_1']= pd.to_datetime(tweets['Date_1']).dt.date
```

In [71]:
```
tweets.head()
```

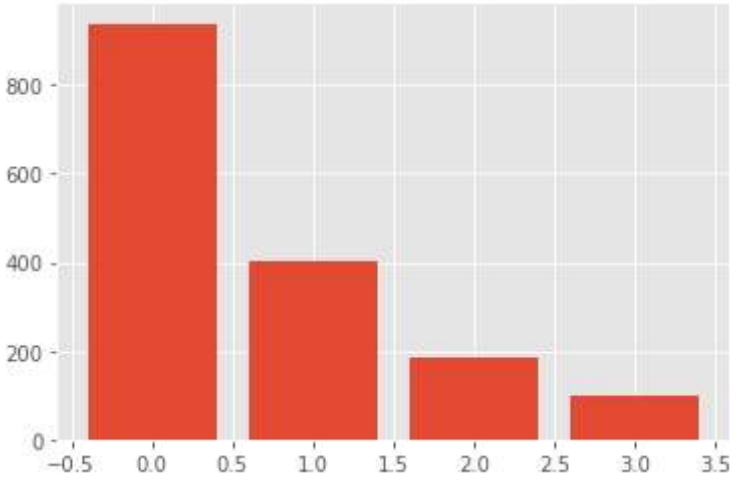
Out[71]:

| | Tweet_Id | Text | Name | Screen_Name | UTC |
|---|---|---|---|---|---|
| 0 | 1597027289266475008 | SEO 2021 Learn Search Engine Optimization With... | Hadley_EichmannPic | eichmannpic | 2022-11 28T00:39:11.000 |
| 1 | 1597026923854454785 | What is Search Engine Optimization, and how ma... | ABDUL KHALAKE IBNA AHMMED | sohelahmmeddm | 2022-11 28T00:37:44.000 |
| 2 | 1597025184619249664 | RT @CdkLabs : Investing in search engine optim... | Hello ✌️❤️ | CasualViewer123 | 2022-11 28T00:30:49.000 |
| 3 | 1597020230886330369 | With over twenty years of experience designing... | Search Engine Social Media Agency Bristol | capidhouser | 2022-11 28T00:11:08.000 |
| 4 | 1597019198755983360 | Looking for a Jr. Search Engine Optimization S... | Tampa SEO Academy | TampaSEOAcademy | 2022-11 28T00:07:02.000 |

In [100]:
```
def get_tweets(row):
    Date_1 = []
    Text = row["Text"].lower()
    if "seo" in Text:
        Date_1.append("seo")
    if "search engine optimization" in Text:
        Date_1.append("search engine optimization")
    return ",".join(Date_1)
tweets["Date_1"] = tweets.apply(get_tweets,axis=1)
```

In [102]:
```
#Plotting Twitter data (Text vs Date)

counts = tweets["Date_1"].value_counts()
plt.bar(range(len(counts)), counts)
plt.show()
print(counts)
```




```
seo,search engine optimization    939
search engine optimization        403
seo                               185
                                   99
Name: Date_1, dtype: int64
```

In [44]:
```
# Filepath to data set
excel_file_path_1 = 'C:\\Users\\saiei\\Downloads\\vicinitas_search_results.xl


tweets_df = pd.read_excel(excel_file_path_1, sheet_name = 'tweets')


print("Number of claims in file:",len(tweets_df))


print(tweets_df.columns)


print(tweets_df.dtypes)
```

```
Number of claims in file: 1626
Index(['Tweet Id', 'Text', 'Name', 'Screen Name', 'UTC', 'Created At',
       'Favorites', 'Retweets', 'Language', 'Client', 'Tweet Type', 'URLs',
       'Hashtags', 'Mentions', 'Media Type', 'Media URLs', 'Unnamed: 16',
       'Unnamed: 17', 'Unnamed: 18'],
      dtype='object')
Tweet Id        int64
Text           object
Name           object
Screen Name    object
UTC            object
Created At     object
Favorites       int64
Retweets        int64
Language       object
Client         object
Tweet Type     object
URLs           object
Hashtags        int64
Mentions        int64
Media Type     object
Media URLs     object
Unnamed: 16    object
Unnamed: 17    object
Unnamed: 18    object
dtype: object
```

In [45]:
```
    # Extracting unstructured date and converting to datetime
tweets_df['Date_2']= pd.to_datetime(tweets_df['Created At'], errors='coerce')


tweets_df['Date_2']= pd.to_datetime(tweets_df['Date_2']).dt.date
```

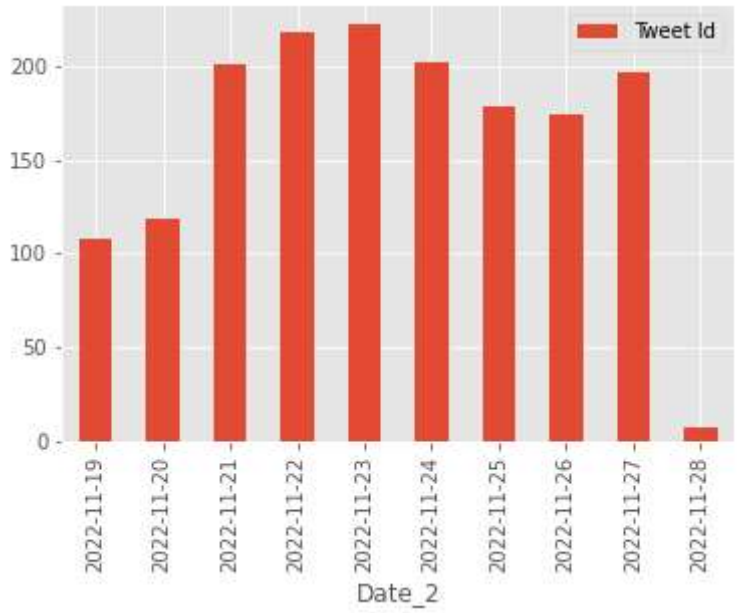
In [46]:
```
#plotting Twitter data (Tweet id vs Date)

df_1_freq = df_1.groupby(['Date_2']).count()


#df_1_freq = df_1_plot[['Tweet Id']]

df_1_freq.plot.bar(y='Tweet Id', use_index=True)
```

Out[46]: <AxesSubplot:xlabel='Date_2'>




In [49]:
```
from sklearn.metrics import accuracy_score

print('Accuracy: %.3f' % accuracy_score(testY, predY))
```

```
Accuracy: 0.000
```

In [61]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
import pandas as pd
from sklearn.metrics import accuracy_score
```

In [ ]:

```
features = tweets.drop(float("Tweet_Id"), axis=1)
targets = tweets[float("Tweet_Id")]

train_features, test_features, train_targets, test_targets = train_test_split

# Train the model
tree = DecisionTreeClassifier(criterion="gini", min_samples_leaf = 4, max_dep
tree = tree.fit(train_features, train_targets)

# Predict the classes of new, unseen data
prediction = tree.predict(test_features)

# Check the accuracy
score = tree.score(test_features, test_targets)
print("The prediction accuracy is: {:0.2f}%".format(score * 100))
```