data_process

2023 年 2 月 17 日

1 demo

1.1 数据预处理 Problem_C_Data_Wordle.xlsx

1. 读取数据
2. 数据清洗
3. 数据分析

[32]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

1. 读取数据

[33]:
```
# 1.1 读取数据
df = pd.read_excel('Problem_C_Data_Wordle.xlsx')
```

[34]:
```
# 将第一行作为列名
df.columns = df.iloc[0]
# 删除第一行
df = df.drop(0)
# 删除空列
df = df.dropna(axis=1, how='all')
# 重置索引
df = df.reset_index(drop=True)
# 1.2 查看数据
df.head()
```

[34]:

0	Date	Contest number	Word	Number of reported results	\
0	2022-12-31 00:00:00	560	manly	20380	
1	2022-12-30 00:00:00	559	molar	21204	

```
2  2022-12-29 00:00:00            558  havoc                       20001
3  2022-12-28 00:00:00            557  impel                       20160
4  2022-12-27 00:00:00            556  condo                       20879

0 Number in hard mode 1 try 2 tries 3 tries 4 tries 5 tries 6 tries  \
0                1899     0       2      17      37      29      12
1                1973     0       4      21      38      26       9
2                1919     0       2      16      38      30      12
3                1937     0       3      21      40      25       9
4                2012     0       2      17      35      29      14

0 7 or more tries (X)
0                    2
1                    1
2                    2
3                    1
4                    3
```

2. 数据清洗

[35]:
```
# 2.1 查看数据类型
df.dtypes
```

[35]:
```
0
Date                           object
Contest number                 object
Word                           object
Number of  reported results    object
Number in hard mode            object
1 try                          object
2 tries                        object
3 tries                        object
4 tries                        object
5 tries                        object
6 tries                        object
7 or more tries (X)            object
dtype: object
```

[36]:
```
# 2.2 查看数据缺失情况
df.isnull().sum()
```

[36]:
```
0
Date                            0
Contest number                  0
Word                            0
Number of  reported results     0
Number in hard mode             0
1 try                           0
2 tries                         0
3 tries                         0
4 tries                         0
5 tries                         0
6 tries                         0
7 or more tries (X)             0
dtype: int64
```

[37]:
```
# 2.3 查看数据分布情况
df.describe()
```

[37]:
```
0                     Date  Contest number   Word  \
count                  359             359    359
unique                 359             359    359
top     2022-12-31 00:00:00             560  manly
freq                     1               1      1

0       Number of  reported results  Number in hard mode  1 try  2 tries  \
count                           359                  359    359      359
unique                          357                  344      6       22
top                          218595                10343      0        2
freq                              2                    2    221       56

0       3 tries  4 tries  5 tries  6 tries  7 or more tries (X)
count       359      359      359      359                  359
unique       36       32       31       30                   21
top          16       35       24        9                    1
```

```
freq          19        38        32        30                   146
```

3. 数据分析 我们在下面可以看到字符不够五个问题

分析 Word 列

[38]:
```
df['Word'].value_counts()
```

[38]:
```
manly    1
gamer    1
larva    1
forgo    1
story    1
        ..
inter    1
whoop    1
taunt    1
leery    1
slump    1
Name: Word, Length: 359, dtype: int64
```

[39]:
```
# 查看 Word 的列的字符串长度
df['Word'].str.len().describe()
```

[39]:
```
count    359.000000
mean       5.000000
std        0.105703
min        4.000000
25%        5.000000
50%        5.000000
75%        5.000000
max        6.000000
Name: Word, dtype: float64
```

[40]:
```
# 找出字符串长度不等于 5 的行
df[df['Word'].str.len() != 5]
```

[40]:
```
0                   Date Contest number    Word Number of  reported results  \
15   2022-12-16 00:00:00            545  rprobe                        22853
```

```
35   2022-11-26 00:00:00            525    clen                      26381
246  2022-04-29 00:00:00            314    tash                     106652
353  2022-01-12 00:00:00            207  favor                      137586

0   Number in hard mode 1 try 2 tries 3 tries 4 tries 5 tries 6 tries  \
15                  2160     0       6      24      32      24      11
35                  2424     1      17      36      31      12       3
246                 7001     2      19      34      27      13       4
353                 3073     1       4      15      26      29      21

0   7 or more tries (X)
15                    3
35                    0
246                   1
353                   4
```

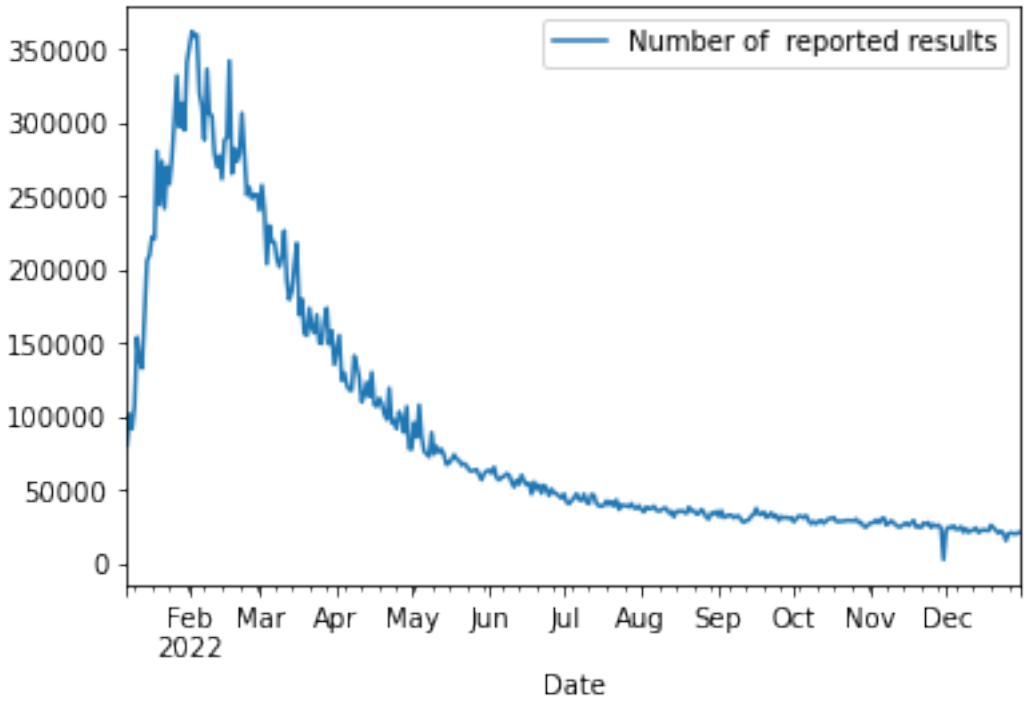
分析 Number of reported results 列

[41]:
```
df['Number of  reported results'].describe()
# count 359 <-- 359 个单词
# unique 357 <-- unique 的意思是不重复的，这里 359 行中有 2 行重复了
# top 218595 <-- top 的意思是出现频率最高的，这里出现频率最高的是 218595
# freq 2
```

[41]:
```
count        359
unique       357
top       218595
freq           2
Name: Number of  reported results, dtype: int64
```

[42]:
```
# 以时间为横坐标，Number of  reported results 为纵坐标，画出折线图
df.plot(x='Date', y='Number of  reported results', kind='line')
```

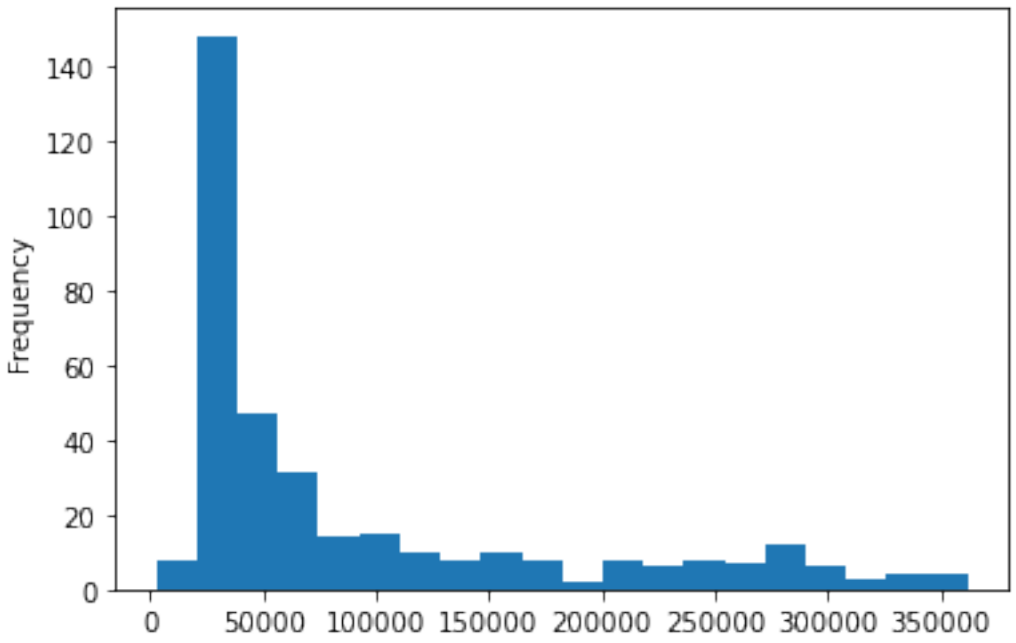
[42]:
```
<AxesSubplot:xlabel='Date'>
```



上图中我们可以发现有一个异常值，我们需要去除，用样条插值法和埃尔米特插值法取平均值，代码如下：

[48]:
```
# 单独取出 Number of  reported results 列
df['Number of  reported results'].plot(kind='hist',bins=20)
```

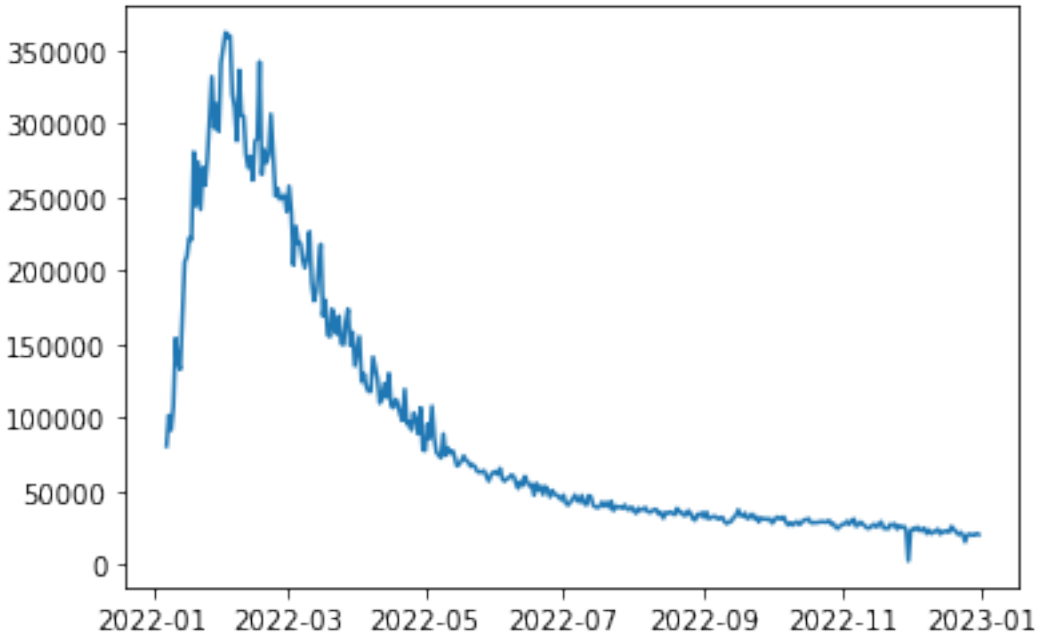
[48]: <AxesSubplot:ylabel='Frequency'>



[49]:
```
df_Number_of_reported_results = df['Number of  reported results']
```

[51]:
```
# 用 plt 画图，横坐标为 df 的 Date 列，纵坐标为 df_Number_of_reported_results 列
plt.plot(df['Date'], df_Number_of_reported_results)
```

[51]: [<matplotlib.lines.Line2D at 0x155d8959490>]



[52]:
```
print(df['Number of  reported results'].isnull().sum())
```

```
0
```

[54]:
```
df_Number_of_reported_results[df_Number_of_reported_results<10000]
```

[54]:
```
31    2569
Name: Number of  reported results, dtype: object
```

[55]:
```
error_index =␣
↪df_Number_of_reported_results[df_Number_of_reported_results<10000].index
```

[58]:
```
error_index
```

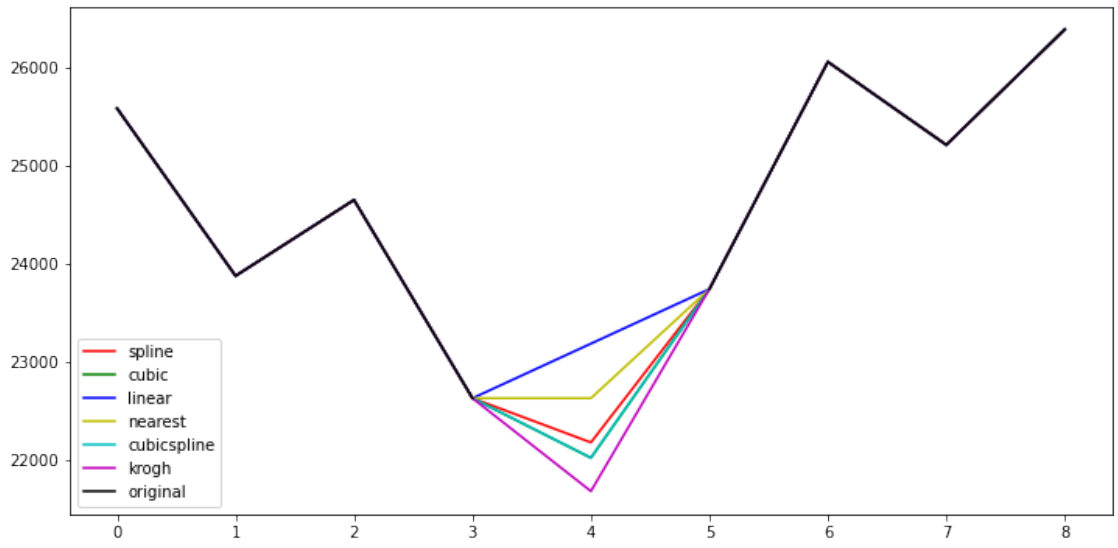
[58]:
```
Int64Index([31], dtype='int64')
```

[67]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data = np.array([25577, 23873, 24646, 22628, np.nan, 23739, 26051, 25206,␣
↪26381])
```

```
spline = pd.Series(data).interpolate(method='spline', order=2) # 用 2 阶样条插值法填补缺失值
cubic = pd.Series(data).interpolate(method='cubic') # 用 3 阶样条插值法填补缺失值
linear = pd.Series(data).interpolate(method='linear') # 用线性插值法填补缺失值
nearest = pd.Series(data).interpolate(method='nearest') # 用最近邻插值法填补缺失值
krogh = pd.Series(data).interpolate(method='krogh') # 用 Krogh 插值法填补缺失值␣
↪krogh 就是 hermite 插值法

plt.figure(figsize=(12, 6))
plt.plot(spline, 'r', label='spline')
plt.plot(cubic, 'g', label='cubic')
plt.plot(linear, 'b', label='linear')
plt.plot(nearest, 'y', label='nearest')
plt.plot(cubicspline, 'c', label='cubicspline')
plt.plot(krogh, 'm', label='krogh')
plt.plot(data, 'k', label='original')
plt.legend(loc='best')
plt.show()
```



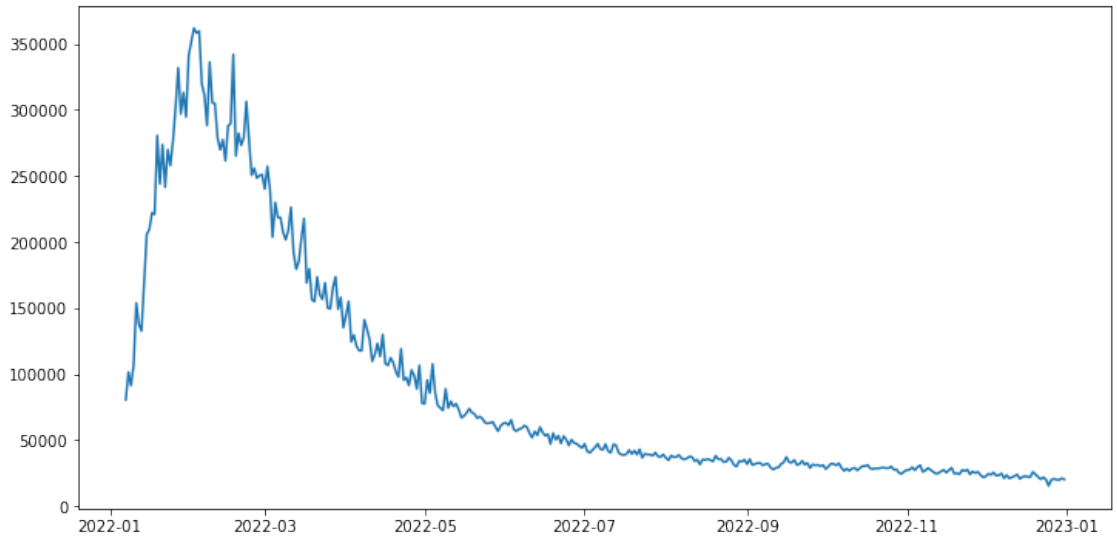
[72]:
```
# 我们采用 spline 和 krogh 的均值
df_Number_of_reported_results[error_index] = int((spline[4] + krogh[4])/2)
```

[73]:
```
df_Number_of_reported_results[error_index]
```

[73]:
```
31    21929
Name: Number of  reported results, dtype: object
```

[77]:
```
plt.figure(figsize=(12, 6))
plt.plot(df['Date'], df_Number_of_reported_results)
```

[77]:
```
[<matplotlib.lines.Line2D at 0x155da380af0>]
```



[75]:
```
# 保存好数据修改的 df_Number_of_reported_results
df_Number_of_reported_results.to_csv('df_Number_of_reported_results.csv',␣
↪index=False)
```

下一节用 lstm 预测 Number of reported results 列