In [1]:
```
#NLP model for sabic WO notes
```

In [362]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [363]:
```
#using an excel file I made while at SABIC with lots of PM info, everything e
#we will pretend the notes column that I used while working on this at sabic
```

In [364]:
```
excel = "C:\\Users\\brian\\Documents\\Data Sabic\\PM_OP_example.xlsx"
```

In [365]:
```
df = pd.read_excel(excel)
```

In [366]:
```
pd.options.display.float_format = '{:20.0f}'.format
```

In [367]:
```
df.head()
```

Out[367]:

	MaintenancePlan	MaintItem text	Cycle	Unit	ABC	Notes	Sched.StartDate	Equipment	So fiel
0	BL1000001830	SITE WIDE CHAINFALL INSPECTION	3	MON	E	no action	2019-03-11 00:00:00	NaN	Na
1	BL1000001831	SITE WIDE LEVER HOIST INSPECTION	3	MON	E	no action	2019-03-11 00:00:00	NaN	Na
2	BL1000001899	SAN CALDWELL SLINGS INSPECTION	13	WK	E	no action	2019-03-11 00:00:00	NaN	Na
3	BL1000001863	FIREALARM ZONE # 19 AUDIBLE ALARM &	19	WK	E	no action	2019-03-11 00:00:00	NaN	Na
4	BL1000001233	LN1, 2 VENTILATION FANS ROOF TOP EXHAUST	26	WK	E	delete	2019-03-11 00:00:00	NaN	Na

In [368]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2006 entries, 0 to 2005
Data columns (total 12 columns):
MaintenancePlan    2005 non-null object
MaintItem text     2005 non-null object
Cycle              2000 non-null float64
Unit               2000 non-null object
ABC                2004 non-null object
Notes              1796 non-null object
Sched.StartDate    2005 non-null object
Equipment          1581 non-null object
Sort field         1581 non-null object
Functional Loc.    2001 non-null object
Work center        2005 non-null object
Last order         1835 non-null float64
dtypes: float64(2), object(10)
memory usage: 188.1+ KB
```

In [369]:
```
df['Sort field'].fillna(value='0 NA',inplace=True)
```

In [370]:
```
df['Equipment'].fillna(value='NA',inplace=True)
```

In [371]:
```
df['MaintenancePlan'].fillna(value='NA',inplace=True)
```

In [372]:
```
df['MaintItem text'].fillna(value='NA',inplace=True)
```

In [373]:
```
df['Cycle'].fillna(value=0,inplace=True)
```

In [374]:
```
df['Unit'].fillna(value='NA',inplace=True)
```

In [375]:
```
df['ABC'].fillna(value='NA',inplace=True)
```

In [376]:
```
df['Notes'].fillna(value='None',inplace=True)
```

In [377]:
```
df['Sched.StartDate'].fillna(value='NA',inplace=True)
```

In [378]:
```
df.iloc[363]
```

Out[378]:
```
MaintenancePlan                    NA
MaintItem text                     NA
Cycle                               0
Unit                               NA
ABC                                NA
Notes                            None
Sched.StartDate                    NA
Equipment                          NA
Sort field                       0 NA
Functional Loc.                   NaN
Work center                       NaN
Last order                        NaN
Name: 363, dtype: object
```

In [379]:
```
df.drop(363,inplace=True)
```

In [380]:
```
df['Functional Loc.'].fillna(value='NA',inplace=True)
```

In [381]:
```
df['Last order'].fillna(value=0,inplace=True)
```

In [382]:
```
df.isnull().any(axis=0)
```

Out[382]:
```
MaintenancePlan    False
MaintItem text     False
Cycle              False
Unit               False
ABC                False
Notes              False
Sched.StartDate    False
Equipment          False
Sort field         False
Functional Loc.    False
Work center        False
Last order         False
dtype: bool
```

In [383]:
```
df[10:15]
```

Out[383]:

	MaintenancePlan	MaintItem text	Cycle	Unit	ABC	Notes	Sched.StartDate	Equipn
10	BL1000001751	BREATHING AIR FILTER/MONITOR	4	WK	C	delete	2019-03-11 00:00:00	BL000000 33
11	BL1000001753	TRACTOR, MAINTENANCE UTILITY	4	WK	C	six months	2019-03-11 00:00:00	BL000000 53
12	BL1000001777	TRACKMOBILE RENTAL WEEKLY PM FILE #3397	1	WK	C	monthly	2019-03-11 00:00:00	BL000000 53
13	BL1000001779	HERCULES TRACKMOBILE - 020-53-114	1	WK	C	monthly	2019-03-11 00:00:00	BL000000 53
14	BL1000000318	VAT HP-5890II GAS CHROMAT - SAN LAB	1	WK	C	no action	2019-03-11 00:00:00	BL000000 18

In [384]:
```
df['Tech ID'] = df['Sort field'].apply(lambda x: x.split()[0])
```

In [385]:
```
df['Equip Descrip'] = df['Sort field'].apply(lambda x: x.split()[1:])
```

In [386]:
```
df.head()
```

Out[386]:

	MaintenancePlan	MaintItem text	Cycle	Unit	ABC	Notes	Sched.StartDate	Equipment	So fiel
0	BL1000001830	SITE WIDE CHAINFALL INSPECTION	3	MON	E	no action	2019-03-11 00:00:00	NA	N
1	BL1000001831	SITE WIDE LEVER HOIST INSPECTION	3	MON	E	no action	2019-03-11 00:00:00	NA	N
2	BL1000001899	SAN CALDWELL SLINGS INSPECTION	13	WK	E	no action	2019-03-11 00:00:00	NA	N
3	BL1000001863	FIREALARM ZONE # 19 AUDIBLE ALARM &	19	WK	E	no action	2019-03-11 00:00:00	NA	N
4	BL1000001233	LN1, 2 VENTILATION FANS ROOF TOP EXHAUST	26	WK	E	delete	2019-03-11 00:00:00	NA	N

In [415]:
```
z = df['Equip Descrip'][10][0] +' '+ df['Equip Descrip'][10][1]
```

In [403]:
```
z
```

Out[403]:
```
'BREATHING AI'
```

In [399]:
```
def equip(y):
    Describ = y[0]
    lens = y[1]

    if lens == 0:
        return Describ[y][0]
    elif lens == 1:
        return Describ[y][0] + ' ' + Describ[y][1]
    elif lens == 2:
        return Describ[y][0] + ' ' + Describ[y][1] + ' ' + Describ[y][2]
    elif lens == 3:
        return Describ[y][0] + ' ' + Describ[y][1] + ' ' + Describ[y][2] + '
    elif lens == 4:
        return Describ[y][0] + ' ' + Describ[y][1] + ' ' + Describ[y][2] + '
    else:
        return Describ[y][0] + ' ' + Describ[y][1] + ' ' + Describ[y][2] + '
```

In [400]:
```
x['des'] = x[['Equip Descrip','lens']].apply(equip,axis=1)
```

...

In [392]:
```
x['Equip Descrip'][0][0]
```

Out[392]: 'NA'

In [390]:
```
x[['Equip Descrip','lens']].head()
```

Out[390]:

	Equip Descrip	lens
0	[NA]	1
1	[NA]	1
2	[NA]	1
3	[NA]	1
4	[NA]	1

In [387]:
```
x = df[['Equip Descrip']]
```

In [388]:
```
x['lens'] = x['Equip Descrip'].apply(len)
```

```
C:\Users\brian\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: Setting
WithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/
stable/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pa
ndas-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

In [389]:
```
x[10:15]
```

Out[389]:

	Equip Descrip	lens
10	[BREATHING, AI]	2
11	[UTLTY, TRACTO]	2
12	[TRCKMBL, RENT]	2
13	[TRCKMBLHERCU]	1
14	[CHROMATOGRAP]	1

In [393]:
```
x['lens'].max()
```

Out[393]: 5

In [242]:
```
len(df.loc[10,'Equip Descrip'])
```

Out[242]: 2