EDA Process of CGC data

October 1, 2019

1 Loading Libaries

[57]:

```
#Common imports
import sys
import os
# Numpy for stastics
import numpy as np

#Python Data Analysis Library
import pandas as pd

#Data visualization
%matplotlib inline
#sets the backend of matplotlib to the 'inline' backend
#%matplotlib notebook
import matplotlib
import seaborn as sns
import matplotlib.pyplot as plt

# Ignore useless warnings (see SciPy issue #5998)
import warnings
warnings.filterwarnings(action="ignore", message="^internal gelsd")
```

2 Data import and cleaning

2.1 Data Import

[72]:

```
df = pd.read_excel('E:/nikhitha/datasets/CGC/CGC Total Power comsumption.xlsx',␣
 ↪sheetname='Sheet1') #loaded or readed the data
```

2.1.1 View the Dataset

df.head()

df.info()

2.2 Removing the Unnecessary columns

[75]:
```
df1 =df.drop(columns = ["UOM","Type of Data", "Tag"]) # drop the unnecessary␣
 ↪columns
```

2.3 Finding Dimensions of data

df1.shape

This data consisting of 37 rows and 1181 columns,which is unstructed form, so we need to make them into structed form by converting rows to columns and vice versa

3 Transpose of data

df1 = df1.T.reset_index() # transpose the dataset converting rows to colmns and columns to rows df1.head()

### Assigning the first row of the dataset to Column header

header = df1.iloc[0] # assigning the first row of the dataset to header header

3.0.1 Loading the data of all rows and columns

df1 = df1[1:] # loading the data of all rows and columns df1.head()

3.0.2 Assigning the column names to the transposed data set

df1 =df1.rename(columns = header) # Assigning the column names to the transposed data set df1.head()

4 Renaming the Column names

[81]:
```
df1=df1.rename(columns={"Description": "DateTime"}) # renamimg the columns names
```

[82]:
```
df1.columns = df1.columns.str.replace(' ', '')
```

[83]:
```
df1.columns = df1.columns.str.replace('-', '')
```

4.0.1 View the Column names

df1.columns

4.0.2 Finding the Structure of data after transpose

df1. shape

After Transpose od data, we got 1180 rows and 38 columns

4.1 Finding the datatype of variables

df1.dtypes

4.2 Datatype Conversion

[87]:

```
a =␣
 ↪('1stStageSuctionTemperature','1stStageSuctionPressure','1stStageDischargeTemperature','1st
 ↪'4thStageDischargePressure','5thStageSuctionTemperature','5thStageSuctionPressure','5thStag
 ↪'3rdStageDischargeFlow','5thStageDischargeFlow','C3SplitterPurgeto4thStageSuction','C2Split
 ↪'HPSteamExtractionPressure','HPSteamExtractionTemperature','E24PGInletTemperature','TotalPo
for i in a:
   # df1[i] =  df1[i].astype(float)
    df1[i] = pd.to_numeric(df1[i], errors = " Coerce")
```

[88]:

```
# Converting into datetime format
df1['DateTime'] =  pd.to_datetime(df1['DateTime'], format='%Y%m%d:%H:%M:%S.%f')
```

4.3 Numerical variables

num_cols = df1._get_numeric_data().columns #finding the numerical data types in the dataset num_cols

4.3.1 Getting Datatypes of variables

df1.dtypes

4.4 Finding the Missing Values

df1.isnull().sum()

CWSupplyTemperature:126,CWFlowtoOlefins:61,CWPressure:65,CompressorSpeed:9,UHPSteamFlowtoKT1:20,U

4.5 Drop the NA values

[92]:

```
df1 = df1.dropna()
```

df1.isnull().sum()

4.6 Descriptive statistics

df1.describe()