**Чепоков Елизар ПИ-18-2**

1. **Краткая информация о проекте**. Технологический процесс добычи нефти. В работе находятся несколько скважин, каждая из которых оснащена насосным оборудованием. Оборудование может работать в различных режимах. Основные технологические параметры о работе оборудования измеряются с дискретностью 1 сутки (раз в 24 часа). В связи с нестабильной связью, особенностями сбора данных (ручной ввод) имеются значительные пропуски в данных
2. **Исходные данные** - файл Excel с параметрами скважин, результатами измерений, режимами работы оборудования. Каждая вкладка файла содержит данные по 1 месяцу работы всего массива скважин.
3. “**Шпаргалка**” для анализа данных

* Обработка данных для исследований.py - реализует парсинг данных из исходного файла и формирование датафрейма. Формируется новый датафрейм с данными только по одной выбранной скважине. Также в файле реализуется начальная очистка данных (удаление столбцов с пустыми значениями во всех строках) для выбранной скважины
* Сохранение\_данных\_в\_CSV\_для\_Influx.py - реализует парсинг данных из исходного файла, подготовку датафрейма для экспорта и сам экспорт в csv файл. Далее можно использоваться скрипт CSV2Influx.py для экспорта в БД InfluxDB

**НИС №7 “Работа по проекту анализа данных IoT в группах”**

# -\*- coding: utf-8 -\*-

"""

Created on Tue May 7 22:29:48 2019

@author: пк

"""

# в консоли IPython -> !pip install influxdb

from influxdb import InfluxDBClient

import pandas as pd

import numpy as np

data = 'Дата замера'

hole = 'Скважина'

x1 = 'Способ эксплуатации'

x2 = 'Режим'

y1 = 'Рпр(ТМ)'

y2 = 'Рзаб(Рпр)'

y3 = 'Рзаб(Нд)'

y4 = 'Рзаб(иссл)'

file\_name = 'Данные для исследований.xlsx'

def read\_all\_sheets(file\_name\_excel):

df = pd.DataFrame()

xls = pd.ExcelFile(file\_name\_excel)

for list\_excel in xls.sheet\_names:

df = df.append(pd.read\_excel(xls, list\_excel, parse\_dates=[data], index\_col=data))

return df

df = read\_all\_sheets(file\_name)

df.sort\_index(inplace=True)

def replace\_text\_values\_in\_x(df, nameX):

dict\_changes = {}

\_list = pd.unique(df[nameX]).tolist()

i = 1

for value in \_list:

if (str(value) != str(np.NaN)):

df.loc[df[nameX] == value, nameX] = i

dict\_changes[i] = value

i += 1

df[nameX] = df[nameX].fillna(len(\_list))

dict\_changes[len(\_list)] = np.NaN

return dict\_changes

what\_replaced\_x1 = replace\_text\_values\_in\_x(df, x1)

what\_replaced\_x2 = replace\_text\_values\_in\_x(df, x2)

all\_data = df.copy()

cleaning\_map = lambda x: str(x).strip()

all\_data[hole] = all\_data[hole].map(cleaning\_map)

all\_df\_to\_influx = all\_data.copy()[[hole, x1, x2, y1, y2, y3, y4]]

influx\_file\_name = '\_data\_2018\_01-07'

list\_of\_holes = pd.unique(all\_data[hole]).tolist()

count\_empty\_data = 0

list\_empty\_data = []

cleaned\_data = pd.DataFrame()

for \_hole in list\_of\_holes:

df\_to\_influx = all\_df\_to\_influx[all\_df\_to\_influx[hole] == \_hole][[x1, x2, y1, y2, y3, y4]]

df\_to\_influx.insert(loc=0, column='Время', value=df\_to\_influx.index.time[0])

temp\_df = df\_to\_influx[[y1, y2, y3, y4]].dropna(axis=0, how='all')

if (not temp\_df.empty):

cleaned\_data = cleaned\_data.append(df\_to\_influx)

df\_to\_influx.to\_csv('output\\' + str(\_hole) + influx\_file\_name+'.csv', encoding='cp1251', sep=';')

else:

#print('Для скважины '+ \_hole +' нет данных по давлению!')

list\_empty\_data.append(\_hole)

count\_empty\_data += 1

print('Количество скважен без данных по давлению: '+ str(count\_empty\_data))

print('Список скважен без данных по давлению:')

i=1

for \_hole in list\_empty\_data:

print(str(i)+'. '+\_hole)

i+=1

print('Сохранение в отдельные файлы ".csv" выполнено!')

"""

writer = pd.ExcelWriter(influx\_file\_name+'.xlsx', engine='xlsxwriter', datetime\_format='dd.mm.yyyy', date\_format='dd.mm.yyyy')

df\_to\_influx.to\_excel(writer)

writer.save()

ex = df\_to\_influx.dropna()

writer = pd.ExcelWriter(influx\_file\_name+'1.xlsx', engine='xlsxwriter', datetime\_format='dd.mm.yyyy', date\_format='dd.mm.yyyy')

ex.to\_excel(writer)

writer.save()

"""

#ex1 = all\_data.copy()[['Скважина', 'Режим', 'Способ эксплуатации', 'Рзаб(иссл)']].dropna()

# making connection to influx

#client = InfluxDBClient(host='localhost', port=8086, username='myuser', password='mypass' ssl=True, verify\_ssl=True)

# -\*- coding: utf-8 cp1251 -\*-

"""

Created on 0 0:00:00 0000

@author: Bloodies

"""

# !pip install influxdb

from influxdb import InfluxDBClient

import pandas as pd

import numpy as np

data = 'Дата замера'

hole = 'Скважина'

x1 = 'Способ эксплуатации'

x2 = 'Режим'

y1 = 'Рпр(ТМ)'

y2 = 'Рзаб(Рпр)'

y3 = 'Рзаб(Нд)'

y4 = 'Рзаб(иссл)'

file\_name = 'data.xlsx'

def read\_all\_sheets(file\_name\_excel):

df = pd.DataFrame()

xls = pd.ExcelFile(file\_name\_excel)

for list\_excel in xls.sheet\_names:

df = df.append(pd.read\_excel(xls, list\_excel, parse\_dates=[data], index\_col=data))

return df

df = read\_all\_sheets(file\_name)

df.sort\_index(inplace=True)

def replace\_text\_values\_in\_x(df, nameX):

dict\_changes = {}

\_list = pd.unique(df[nameX]).tolist()

i = 1

for value in \_list:

if (str(value) != str(np.NaN)):

df.loc[df[nameX] == value, nameX] = i

dict\_changes[i] = value

i += 1

df[nameX] = df[nameX].fillna(len(\_list))

dict\_changes[len(\_list)] = np.NaN

return dict\_changes

what\_replaced\_x1 = replace\_text\_values\_in\_x(df, x1)

what\_replaced\_x2 = replace\_text\_values\_in\_x(df, x2)

all\_data = df.copy()

cleaning\_map = lambda x: str(x).strip()

all\_data[hole] = all\_data[hole].map(cleaning\_map)

all\_df\_to\_influx = all\_data.copy()[[hole, x1, x2, y1, y2, y3, y4]]

influx\_file\_name = '\_data\_2018\_01-07'

list\_of\_holes = pd.unique(all\_data[hole]).tolist()

count\_empty\_data = 0

fill\_method = 'bfill'

list\_empty\_data = []

cleaned\_data = pd.DataFrame()

for \_hole in list\_of\_holes[1:100]:

if count\_empty\_data > 0:

fill\_method = 'ffill'

df\_to\_influx = all\_df\_to\_influx[all\_df\_to\_influx[hole] == \_hole][[x1, x2, y1, y2, y3, y4]]

df\_to\_influx.insert(loc=0, column='Время', value=df\_to\_influx.index.time[0])

df\_to\_influx.insert(loc=0, column='Скважина', value=\_hole)

temp\_df = df\_to\_influx[[y1, y2, y3, y4]].dropna(axis=1, how='all')

if (not temp\_df.empty):

cleaned\_data = cleaned\_data.append(df\_to\_influx)

#cleaned\_data[y1].fillna(method=fill\_method, inplace=True)

#cleaned\_data[y2].fillna(method=fill\_method, inplace=True)

#cleaned\_data[y2].fillna(method=fill\_method, inplace=True)

#cleaned\_data[y3].fillna(method=fill\_method, inplace=True)

#cleaned\_data[y4].fillna(method=fill\_method, inplace=True)

cleaned\_data.to\_csv('output\\' + 'holes'+'.csv', encoding='cp1251', sep=';')

else:

list\_empty\_data.append(\_hole)

count\_empty\_data += 1

print('Сохранение в файл ".csv" выполнено!')