Handlind Missing Data

Date	19 September 2022
Team ID	PNT2022TMID25121
Project Name	Project - Crude Oil Price Prediction
Maximum Marks	4 Marks

Handlind Missing Data

Oil field data are not always accurate and complete. According to Nobakht et al. (2009), corrupted and missing data costs the industry $60 billion annually. Unless extreme caution is taken to collect data, every dataset is expected to have at least 1-5% error from different sources i.e. human error, measurement setup error or measurement equipment malfunction. The main complication for dealing with missing and corrupted data is how far we understand our data behavior. More understanding means better decisions will be made to remedy missing and corrupted data. There are generally two methods used to deal with the missing data: 1- Drop the missing intervals; 2- Estimate the expected value for the missing point. In this paper both methods will be tested and applied on missing production and injection rates in waterfloods projects. A simple reservoir model was used to calculate the expected value for the missing values of the production rates, namely the Resistivity Model (RM) by Albertoni (2003). Reverse modelling was utilized to estimate the missing values for injection rates. Several cases were simulated with two missing ratios: low (15%) and high (30%) in both the production and injection data. Missing points were generated in the datasets in the form of four patterns (Arbitrary, Monotone, Multivariate and Modified Multivariate). The missing data locations were selected in a Monte Carlo-like manner and results were averaged from 400 realizations for each pattern.

HANDLEING MISSING DATA

In [1]:

```
import pandas as pd
ds=pd.read_excel(r"C:\Users\Dhyalan\Desktop\Crude Oil Prices Daily1.xlsx")
ds.shape
```

Out[1]:

```
(8223, 2)
```

In [2]:

```
ds.head()
```

Out[2]:

	Date	Closing Value
0	1986-01-02	25.56
1	1986-01-03	26.00
2	1986-01-06	26.53
3	1986-01-07	25.85
4	1986-01-08	25.87

In [8]:

```
ds.isnull().sum()
```

Out[8]:

```
Date             0
Closing Value    7
dtype: int64
```

In [11]:

```
hd=ds.dropna()
print(hd)
```

```
           Date  Closing Value
0    1986-01-02          25.56
1    1986-01-03          26.00
2    1986-01-06          26.53
3    1986-01-07          25.85
4    1986-01-08          25.87
...         ...            ...
8217 2018-07-02          73.89
8218 2018-07-03          74.19
8220 2018-07-05          73.05
8221 2018-07-06          73.78
8222 2018-07-09          73.93

[8216 rows x 2 columns]
```

In [12]:

```
hd.isnull().sum()
```

Out[12]:

```
Date            0
Closing Value   0
dtype: int64
```

In [ ]:

```

```