Capstone Project 1: Data Wrangling

Appliances Energy Prediction –

Business Problem Description – Dataset contains the house temperature and humidity conditions were monitored with a ZigBee wireless sensor network. As per the description on UCI website, each wireless node transmitted the temperature and humidity conditions around 3.3 min, Then, the wireless data was averaged for 10 minutes periods. The energy data was logged every 10 minutes with m-bus energy meters. Combining this data with the weather data based on the date time columns

Link for the Jupyter file -
https://github.com/arijitsinha80/Springboard/blob/master/Project/Capstoneproject_Phase1.ipynb

We have two data sets - **energydata_complete.csv** and **CrudeOilPrice.csv**. We have taken two diffe rent dataset to get better prediction with analyzing the engorge consumed and how was the fuel pric e during the particular date.

```
df = pd.read_csv('energydata_complete.csv')
```

```
df.head(2)
```

	date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	...	T9	RH_9	T_out	Press_
0	2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.7900	19.79	44.73	19.0	...	17.033333	45.53	6.600000	733.5
1	2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.7225	19.79	44.79	19.0	...	17.066667	45.56	6.483333	733.6

2 rows × 29 columns

```
dfoil = pd.read_csv('CrudeOilPrice.csv', infer_datetime_format=True)
```

```
dfoil.head(2)
```

	date	value
0	2/3/10	76.98
1	2/4/10	73.14

We do not have any missing values in energydata_complete.csv; it has 19735 observation with 29 attr ibutes pertaining to temperature, humidity, light, wind speed, dew, and visibility from local weather c hannel.

```
print("Number of instances in dataset = {}".format(df.shape[0]))
print()
print("Total number of columns = {}".format(df.columns.shape[0]))
print()
print("Column wise count of null values:-")
print()
print(df.isnull().sum())
print()
print(df.info())
```

```
Number of instances in dataset = 19735

Total number of columns = 29
```

We do not have any missing value in CrudeOilPrice.csv, which has the fuel price for respective months and dates. This dataset has 2519 observation and 2 attributes of date and fuel price.

```
print(dfoil.shape)
print()
print(dfoil.describe())
print()
print(dfoil.info())
```

```
(2519, 2)
```

1. The dataset is from 2016-01-11 and 2016-05-27; have data starting JAN to MAY of 2016.

```
print("Data Captured for the period of ", min(df.date), 'and ', max(df.date))
```

```
Data Captured for the period of  2016-01-11 17:00:00 and  2016-05-27 18:00:00
```

2. From the above reading of each sensor is between 14.89 and 29.85 but T6 is between -6 and 28.29. The possible reason can be its reading are for outside.

```
print(df[temp_cols].describe())
```

```
                T1             T2             T3             T4             T5  \
count  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000
mean      21.686571     20.341219     22.267611     20.855335     19.592106
std        1.606066      2.192974      2.006111      2.042884      1.844623
min       16.790000     16.100000     17.200000     15.100000     15.330000
25%       20.760000     18.790000     20.790000     19.530000     18.277500
50%       21.600000     20.000000     22.100000     20.666667     19.390000
75%       22.600000     21.500000     23.290000     22.100000     20.619643
max       26.260000     29.856667     29.236000     26.200000     25.795000

                T6             T7             T8             T9
count  19735.000000  19735.000000  19735.000000  19735.000000
mean       7.910939     20.267106     22.029107     19.485828
std        6.090347      2.109993      1.956162      2.014712
min       -6.065000     15.390000     16.306667     14.890000
25%        3.626667     18.700000     20.790000     18.000000
50%        7.300000     20.033333     22.100000     19.390000
75%       11.256000     21.600000     23.390000     20.600000
max       28.290000     26.000000     27.230000     24.500000
```

3. From the above reading of each sensor is between 20.46 to 58.79 but RH_5 and RH_6 has ma x of 96.32 and 99.9

```
print(df[hum_cols].describe())
```

```
              RH_1          RH_2          RH_3          RH_4          RH_5  \
count  19735.000000  19735.000000  19735.000000  19735.000000  19735.000000
mean      40.259739     40.420420     39.242500     39.026904     50.949283
std        3.979299      4.069813      3.254576      4.341321      9.022034
min       27.023333     20.463333     28.766667     27.660000     29.815000
25%       37.333333     37.900000     36.900000     35.530000     45.400000
50%       39.656667     40.500000     38.530000     38.400000     49.090000
75%       43.066667     43.260000     41.760000     42.156667     53.663333
max       63.360000     56.026667     50.163333     51.090000     96.321667

              RH_6          RH_7          RH_8          RH_9
count  19735.000000  19735.000000  19735.000000  19735.000000
mean      54.609083     35.388200     42.936165     41.552401
std       31.149806      5.114208      5.224361      4.151497
min        1.000000     23.200000     29.600000     29.166667
25%       30.025000     31.500000     39.066667     38.500000
50%       55.290000     34.863333     42.375000     40.900000
75%       83.226667     39.000000     46.536000     44.338095
max       99.900000     51.400000     58.780000     53.326667
```

4. The max value is 1080wh, whereas 75% of usage is under 100wh. Some of the appliances has high consumption.

5. If we see the statistics for Appliance Attributes, the minimium value is 10 and max value is 10 80, and the mean is 97.69 and 75% of records are below 100 KWH. This column has outliers a nd we will keep them and check during our modeling.

```
: print(df[tgt].describe())
```

```
         Appliances
count  19735.000000
mean      97.694958
std      102.524891
min       10.000000
25%       50.000000
50%       60.000000
75%      100.000000
max     1080.000000
```

When merging the two datasets, in energydata dataset, date is a timestamp and in crudeoilprice data set, date is a date datatype, so we have to normalize the date, in order for us to merge the two datas ets.

1. After the merge, we observe that " values" columns is merged on the dataset, but it doesnt h ave all the dates values and 5904 records has null values.

```
rv2           19735 non-null float64
 value        13831 non-null float64
dtypes: float64(27), int64(2), object(1)
memory usage: 4.7+ MB
None
```

To solve these null values, we used the forward fill method and value column was populated with previous day values for the records, which were null and renamed the column to "oilprice".

```
1]: # Use forward fill to update the null values
    dfmerge[' value'].fillna(method='ffill', inplace=True)
```

After which, the shape of the merged dataset is as below –

```
print(dfmerge.shape)
print()
print(dfmerge.describe())
print()
print(dfmerge.info())
```

```
(19735, 30)
```

We have saved the dataset as input for other phase of this capstone project.