Task 2 - Exploratory Data Analysis & Data Cleaning

Prepared by: Ali Mohammad Jawad Abdullah

Data Dictionary

Summary

- 5 flag columns that has high amounts of zeros and missing values (> 75% of total)
- 3 columns that are redundant and can be dropped because they have no variance or low variance
- 3 Redundant pairs are also there: Estimated energy and power, wind speed and power, temp and power

Import Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objs as go
import plotly.io as pio
import ydata_profiling as pp
import warnings
warnings.filterwarnings('ignore')

from ydata_profiling import ProfileReport

%matplotlib inline

# Set plotly renderer for better display
pio.renderers.default = 'notebook'

# Set pandas display options
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 100)
pd.set_option('display.float_format', '{:.2f}')

# Set seaborn style
sns.set_style("whitegrid")

# Set matplotlib inline and figure size
plt.rcParams['figure.figure.dpi'] = 100
plt.rcParams['figure.figsize'] = (12, 6)
```

```
Matplotlib building font cache;
delete the directory /home/jovyan/.cache/matplotlib to force a rebuild
```

```
df.head(5),df.tail(5),df.info()
```

```
   id  ...  estimated_power_kw
0   0  ...                0.00
1   1  ...                0.00
2   2  ...                0.00
3   3  ...                0.00
4   4  ...                0.00

[5 rows x 21 columns],
      id  ...  estimated_power_kw
9995  9  ...                0.00
9996  9  ...                0.00
9997  9  ...                0.00
9998  9  ...                0.00
9999  9  ...                0.00

[5 rows x 21 columns],
```

```
# Drop unwanted columns
df.drop(['Unnamed: 0'], inplace=True)
df.info()
```

```
<class'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   object
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64
 4   smoker    1338 non-null   object
 5   region    1338 non-null   object
 6   charges   1338 non-null   float64
dtype: int64, object(3)
memory usage: 73.2+ KB
```

Correlation

Null Ho hypothesis:

- H_0 :
- H_1 :

```
# Correlation between numerical features
df.corr(numeric_only=True)
```

	age	bmi	children	charges
age	1.000000	0.109272	0.042469	0.299308
bmi	0.109272	1.000000	0.012759	0.198341
children	0.042469	0.012759	1.000000	0.067998
charges	0.299308	0.198341	0.067998	1.000000

Data Preprocessing

```
df.head()
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

```
df.describe()
```

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

```
df.shape
```

```
(1338, 7)
```

Treat Missing Values

```
df.isnull().sum()
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

```
df.isnull().sum().sum
```

```
0
```

Treat Duplicate Values

```
df.duplicated(keep='first').sum()
```

```
1
```

```
df.drop_duplicates(inplace=True)
```

```
df.duplicated(keep='first').sum()
```

```
0
```

One-hot encoding

```
df.info()
```

```
<class'pandas.core.frame.DataFrame'>
Int64Index: 1337 entries, 0 to 1337
Data columns (total 7):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1337 non-null   int64
 1   sex       1337 non-null   object
 2   bmi       1337 non-null   float64
 3   children  1337 non-null   int64
 4   smoker    1337 non-null   object
 5   region    1337 non-null   object
 6   charges   1337 non-null   float64
dtype: int64, object(3)
memory usage: 83.6+ KB
```

```
df['sex_cat'] = df['sex'].astype('category')
df['smoker_cat'] = df['smoker'].astype('category')
df['region_cat'] = df['region'].astype('category')
df.info()
```

```
<class'pandas.core.frame.DataFrame'>
Int64Index: 1337 entries, 0 to 1337
Data columns (total 10):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   age         1337 non-null   int64
 1   sex         1337 non-null   object
 2   bmi         1337 non-null   float64
 3   children    1337 non-null   int64
 4   smoker      1337 non-null   object
 5   region      1337 non-null   object
 6   charges     1337 non-null   float64
 7   sex_cat     1337 non-null   category
 8   smoker_cat  1337 non-null   category
 9   region_cat  1337 non-null   category
dtype: int64, object(3)
memory usage: 83.6+ KB
```

```
df.head()
```

	age	sex	bmi	children	smoker	region	charges	sex_cat	smoker_cat	region_cat
0	19	female	27.900	0	yes	southwest	16884.92400	0	1	3
1	18	male	33.770	1	no	southeast	1725.55230	1	0	2
2	28	male	33.000	3	no	southeast	4449.46200	1	0	2
3	33	male	22.705	0	no	northwest	21984.47061	1	0	1
4	32	male	28.880	0	no	northwest	3866.85520	1	0	1

```
df['sex_cat'].unique()
```

```
array[0, 1]
Categories (2, int64): [0, 1]
```

```
df['smoker_cat'].unique()
```

```
array[1, 0]
Categories (2, int64): [0, 1]
```

```
df['region_cat'].unique()
```

```
array[3, 2, 1, 0]
Categories (4, int64): [0, 1, 2, 3]
```

```
df['sex_cat'].value_counts()
```

```
1    676
0    662
Name: sex_cat, dtype: int64
```

```
df['smoker_cat'].value_counts()
```

```
0    1063
1     274
Name: smoker_cat, dtype: int64
```

```
df['region_cat'].value_counts()
```

```
2    324
3    324
1    324
0    325
Name: region_cat, dtype: int64
```

```
df.drop(['sex', 'smoker', 'region'], inplace=True)
```

```
df.head()
```

	age	bmi	children	charges	sex_cat	smoker_cat	region_cat
0	19	27.900	0	16884.92400	0	1	3
1	18	33.770	1	1725.55230	1	0	2
2	28	33.000	3	4449.46200	1	0	2
3	33	22.705	0	21984.47061	1	0	1
4	32	28.880	0	3866.85520	1	0	1

```
df['charges'].unique()
```

```
array[16884.92400,  1725.55230,  4449.46200, ..., 10600.54830,
       2205.98080, 1629.8335 ]
```

```
100 % size = 10 records
```

```
sns.lmplot(data=df, x='bmi', y='charges')
```