Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	Team-739649
Project Title	Predicting the energy output of wind turbine based on weather condition
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>

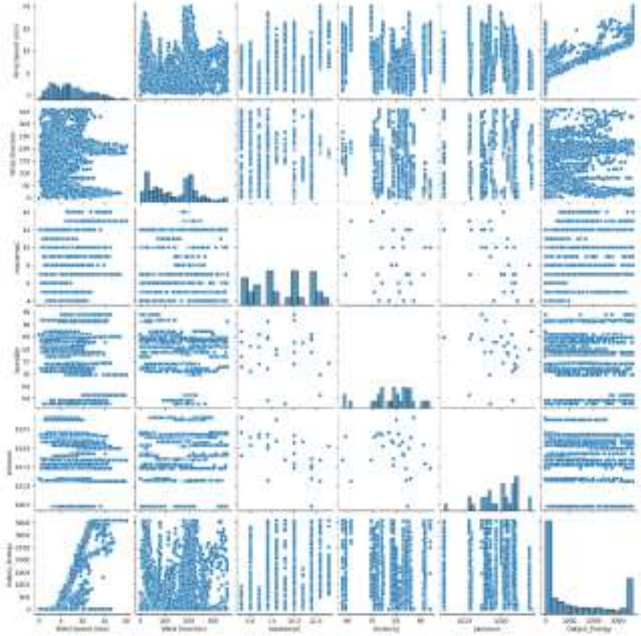
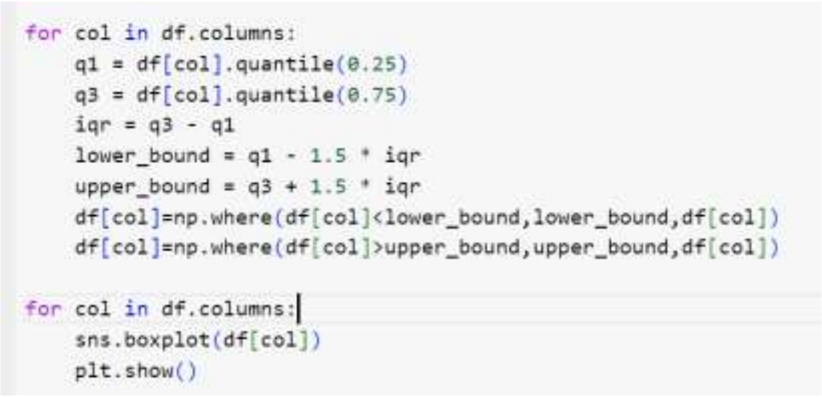
Basic statistics, dimensions, and structure of the data.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4447 entries, 0 to 4446
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Wind Speed (m/s)  4447 non-null   float64
 1   Wind Direction    4447 non-null   float64
 2   maxtempC          4447 non-null   int64  
 3   humidity          4447 non-null   float64
 4   pressure          4447 non-null   float64
 5   Output_Energy     4447 non-null   float64
dtypes: float64(5), int64(1)
```

</td></tr>
<tr><td>Univariate Analysis</td><td>Exploration of individual variables (mean, median, mode, etc.).</td></tr>
</table>

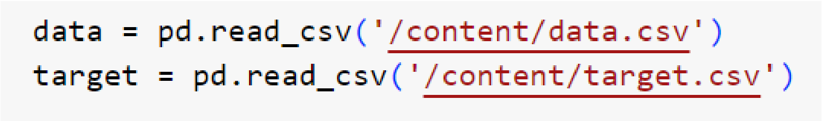
<table>
<tr>
<td></td>
<td>
<table>
<tr><th>Wind Speed (m/s)</th><th>Wind Direction</th><th>maxtempC</th><th>humidity</th><th>pressure</th><th>Output_Ene g</th></tr>
<tr><td><b>count</b></td><td>4447.000000</td><td>4447.000000</td><td>4447.000000</td><td>4447.000000</td><td>4447.00000</td></tr>
<tr><td><b>mean</b></td><td>7.357389</td><td>140.667803</td><td>8.535192</td><td>78.648874</td><td>1019.49165</td></tr>
<tr><td><b>std</b></td><td>4.361162</td><td>93.616266</td><td>3.034301</td><td>9.004574</td><td>5.15432</td></tr>
<tr><td><b>min</b></td><td>0.000000</td><td>0.000000</td><td>4.000000</td><td>54.125000</td><td>1004.54166</td></tr>
<tr><td><b>25%</b></td><td>3.669025</td><td>53.272396</td><td>6.000000</td><td>74.000000</td><td>1015.87500</td></tr>
<tr><td><b>50%</b></td><td>6.717962</td><td>143.424896</td><td>8.000000</td><td>80.041667</td><td>1020.83333</td></tr>
<tr><td><b>75%</b></td><td>10.197950</td><td>206.816154</td><td>12.000000</td><td>84.708333</td><td>1023.45833</td></tr>
<tr><td><b>max</b></td><td>21.621000</td><td>359.942291</td><td>14.000000</td><td>93.958333</td><td>1028.20833</td></tr>
</table>
</td>
</tr>
<tr>
<td>Bivariate Analysis</td>
<td>
Relationships between two variables (correlation, scatter plots).
<table>
<tr><th>Wind Speed (m/s)</th><th>Wind Direction</th><th>maxtempC</th><th>humidity</th><th>pressu</th></tr>
<tr><td><b>Wind Speed (m/s)</b></td><td>1.000000</td><td>0.017336</td><td>0.339107</td><td>0.15185</td></tr>
<tr><td><b>Wind Direction</b></td><td>0.017336</td><td>1.000000</td><td>0.080762</td><td>0.31354</td></tr>
<tr><td><b>maxtempC</b></td><td>0.339107</td><td>0.080762</td><td>1.000000</td><td>0.06532</td></tr>
</table>
</td>
</tr>
</table>

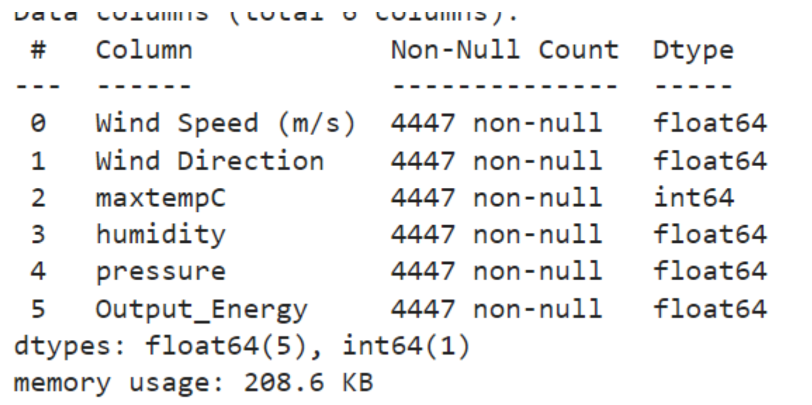
humidity	-0.151853	-0.313542	-0.065329	1.000000	-0.12
pressure	-0.234967	-0.020962	-0.597324	-0.129295	1.00
Output_Energy	0.882457	0.122913	0.403382	-0.251067	-0.24

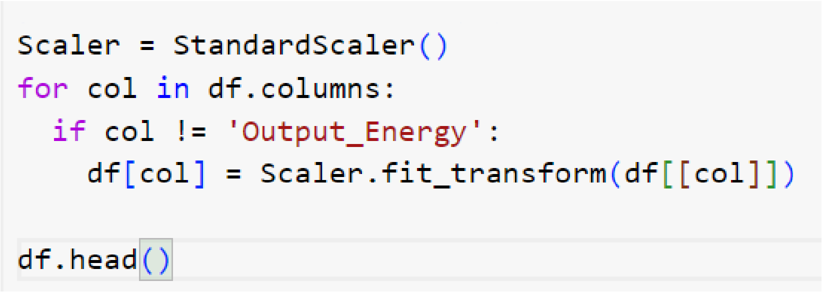
Multivariate Analysis	Patterns and relationships involving multiple variables. 
Outliers and Anomalies	
Data Preprocessing Code Screenshots	

```
for col in df.columns:
    q1 = df[col].quantile(0.25)
    q3 = df[col].quantile(0.75)
    iqr = q3 - q1
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr
    df[col]=np.where(df[col]<lower_bound,lower_bound,df[col])
    df[col]=np.where(df[col]>upper_bound,upper_bound,df[col])

for col in df.columns:
    sns.boxplot(df[col])
    plt.show()
```

Loading Data	
	data = pd.read_csv('/content/data.csv')
	target = pd.read_csv('/content/target.csv')

Handling Missing Data	 Data columns (total 6 columns): # Column Non-Null Count Dtype --- ------ -------------- ----- 0 Wind Speed (m/s) 4447 non-null float64 1 Wind Direction 4447 non-null float64 2 maxtempC 4447 non-null int64 3 humidity 4447 non-null float64 4 pressure 4447 non-null float64 5 Output_Energy 4447 non-null float64 dtypes: float64(5), int64(1) memory usage: 208.6 KB

Data Transformation	
	Scaler = StandardScaler()
	for col in df.columns:
	  if col != 'Output_Energy':
	    df[col] = Scaler.fit_transform(df[[col]])
	
	df.head()

Feature Engineering	Code for creating new features or modifying existing ones.
Save Processed Data	Code to save the cleaned and processed data for future use. df = data