

## Data Collection and Preprocessing Phase

|               |   |
|---------------|---|
| Date          | 11 <sup>th</sup> July 2024  |
| Team ID       | SWTID1720098339   |
| Project Title | Machine learning approach for predicting the price of natural gas |
| Maximum Marks | 6 Marks   |

| Section                | Description  |
|------------------------|--|
| Data Overview          | Mean of natural gas prices over time periods Time series data capturing daily, weekly, or monthly price fluctuations. Tabular format with columns such as date, price, volume, and possibly additional variables like weather data or economic indicators influencing gas prices |
| Univariate Analysis    | Average price of natural gas over a specified period. It is used to fill the missing values.   |
| Bivariate Analysis     | Quantifying the strength and direction of the linear relationship between variables. These tools are used to identify how changes in one variable (e.g., economic factors) affect natural gas prices.  |
| Multivariate Analysis  | Examining how multiple factors like economic indicators, and historical prices interact to influence natural gas prices..  |
| Outliers and Anomalies | Using $r^2$ scores for identification and linegraph for visualisation. Replacing the null values with mean of remaining values in the price.   |
|                        |  |

| Data Preprocessing Code Screenshots |  |
|-------------------------------------|--|
| Loading Data                        | <pre>data = pd.read_csv('daily_csv.csv')</pre>   |
| Handling Missing Data               | <pre>data['Price'] = data['Price'].fillna(data['Price'].mean())</pre>  |
| Data Transformation                 | <pre>scaler = StandardScaler() X_train_scaled = scaler.fit_transform(X_train) X_test_scaled = scaler.transform(X_test)</pre>   |
| Feature Engineering                 | <pre>data['year'] = data['Date'].dt.year data['month'] = data['Date'].dt.month data['day'] = data['Date'].dt.day data['day_of_week'] = data['Date'].dt.dayofweek data['is_weekend'] = data['day_of_week'].apply(lambda x: 1 if x &gt;= 5 else 0)</pre> |
| Save Processed Data                 | <pre>joblib.dump({'model': model, 'scaler': scaler}, 'gas.joblib')</pre>   |