**Data Preprocessing and Exploratory Data Analysis (EDA) Report**

**1. Introduction:** This document outlines the end-to-end data preprocessing and exploratory data analysis (EDA) steps performed on the dataset in preparation for machine learning analysis and decision-making. The preprocessing pipeline aims to clean, transform, and scale the data to ensure that it is suitable for further analysis and modeling.

**2. Initial Data Exploration (EDA):**

We started by performing exploratory data analysis (EDA) to understand the structure of the dataset, the distribution of variables, and the relationships between key features. This process provided insights into the data and helped inform decisions for handling missing values, outliers, and transformations.

**2.1 Dataset Overview:**

* The dataset contains multiple features related to the production process, such as:
  + Material quantities (e.g., COKE\_REQ, SCRAP\_QTY)
  + Energy usage (KWH\_PER\_TON, KWH\_PER\_MIN)
  + Time-related variables (e.g., TAPPING\_TIME, POW\_ON\_TIME, LAB\_REP\_TIME)
  + Chemical compositions (e.g., C, SI, MN)
* Data types include numerical (continuous), categorical, and datetime features.

**2.2 Summary Statistics:**

We performed a summary analysis of the numerical features to understand the distribution of the data. This includes measures such as mean, standard deviation, min, max, and percentiles. The summary statistics helped identify key trends and detect potential issues such as outliers and missing values.

* **Ranges of Key Variables**:
  + COKE\_REQ: 0
  + INJ1\_QTY (Coke Injection Qty): 6.09
  + HOT\_METAL (HOT Metal from MBF): 55.36
  + ENERGY (Energy Consumption): 49078.66
  + MELT\_TIME (Melting Time): 214.48

**2.3 Distribution of Numerical Variables:**

We examined the distribution of key numerical variables to detect skewness, outliers, and normality of the data. Key observations include:

* **Skewness**: Some features, such as INJ1\_QTY and PIGIRON, exhibit high skewness, indicating that they are not normally distributed.
* **Kurtosis**: Certain features, like PIGIRON and GRADE, exhibit high kurtosis, indicating the presence of extreme values (outliers).

**2.4 Visualizing the Data:**

We utilized various visualizations (histograms, box plots, and scatter plots) to gain deeper insights into the distribution and relationships of variables:

* **Histograms** were used to observe the distribution of individual features.
* **Box plots** were employed to visualize the presence of outliers.
* **Correlation heatmaps** were created to identify potential relationships between numerical features.

**3. Data Preprocessing:**

Based on the findings from the EDA, we performed several preprocessing steps to prepare the dataset for modeling.

**3.1 Handling Missing Values:**

Missing values in the dataset were identified and handled as follows:

* LAB\_REP\_TIME: 29 missing values were forward-filled using the previous value (ffill).
* PREV\_TAP\_TIME: 3 missing values were backward-filled using the next available value (bfill).
* Production (MT): 1 missing value was filled with the median value.

python

dataset['LAB\_REP\_TIME'] = dataset['LAB\_REP\_TIME'].ffill()

dataset['PREV\_TAP\_TIME'] = dataset['PREV\_TAP\_TIME'].bfill()

dataset['Production (MT)'] = dataset['Production (MT)'].fillna(dataset['Production (MT)'].median())

**3.2 Feature Engineering:**

Several new features were created to enhance the dataset:

* **Time-Based Features**:
  + Extracted the year, month, and day from the datetime index for time-based analysis.

dataset['year'] = dataset.index.year

dataset['month'] = dataset.index.month

dataset['day'] = dataset.index.day

* **Time Differences**:
  + Calculated the time difference in seconds between important timestamps:
    - pow\_on\_to\_tap\_diff: Time between TAPPING\_TIME and POW\_ON\_TIME.
    - lab\_report\_and\_tap\_diff\_seconds: Time between LAB\_REP\_TIME and PREV\_TAP\_TIME.

dataset['pow\_on\_to\_tap\_diff'] = (dataset['TAPPING\_TIME'] - dataset['POW\_ON\_TIME']).dt.total\_seconds()

dataset['lab\_report\_and\_tap\_diff\_seconds'] = (dataset['LAB\_REP\_TIME'] - dataset['PREV\_TAP\_TIME']).dt.total\_seconds()

**3.3 Removing Duplicate Rows:**

We checked for duplicate rows in the dataset, confirming that no duplicates were present:

dataset.duplicated().sum() # Output: 0

**3.4 Zero Variance Columns:**

Columns with zero variance (constant columns) were identified and removed as they do not provide any useful information for analysis. These included:

* COKE\_REQ, TP, MSTB, SHRAD, REMET, O2REQ, and others.

dataset = dataset.drop(columns=variances[variances == 0].index)

**4. Feature Encoding:**

Categorical variables such as GRADE and SECTION\_IC were one-hot encoded to transform them into numerical representations, which is required for machine learning algorithms. The first category of each feature was dropped to avoid multicollinearity.

dataset = pd.get\_dummies(dataset, columns=['GRADE', 'SECTION\_IC'], drop\_first=True)

**4.1 Data Type Conversion:**

After encoding, the dataset was converted into integers for compatibility with machine learning algorithms.

dataset = dataset.astype(int)

**5. Data Scaling:**

**5.1 Robust Scaling:**

To scale numerical features and handle outliers effectively, we applied **RobustScaler**. This technique scales features based on the interquartile range (IQR) and median, making the data more robust to outliers.

from sklearn.preprocessing import RobustScaler

scaler = RobustScaler()

df\_scaled = scaler.fit\_transform(dataset)

**6. Final Output:**

The preprocessed and scaled dataset was saved into a CSV file (preprocessed\_data.csv) for further analysis, modeling, or reporting.

df\_scaled.to\_csv('preprocessed\_data.csv')

**7. Summary:**

* **Missing Values**: Handled through forward-fill, backward-fill, and median imputation.
* **Feature Engineering**: Time-based features and time differences were added.
* **Data Cleaning**: Duplicates were removed, and features with zero variance were dropped.
* **Feature Encoding**: Categorical features were one-hot encoded and converted into integers.
* **Scaling**: RobustScaler was applied to numerical features to handle outliers.

**8. Next Steps:**

* **Modeling**: The cleaned and scaled data is now ready for machine learning modeling.
* **Performance Metrics**: Future steps include evaluating the performance of various models using this preprocessed data.