Documentation: Overview of the notebook

NOTE:

Other than this file, documentation has been done through

- **Comments, headings and docstrings in the notebook**
- **Readme on Github**

1. Data engineering

- **Data Acquisition**:
    - Imported in-process core densities from SAP through an Excel file.
    - Initialized connection strings and established a connection to the Wonderware server to extract inline density data for both production lines over a 10-minute timeframe.
- **Data Cleaning and Integration**:
    - Cleaned and processed the fetched data, removing nulls and outliers, and computed the average densities per 10-minute interval.
    - Integrated this cleaned inline density data with the in-process density data to create a unified dataset.
- **Additional Datasets**:
    - Created additional datasets, including one that averages inline density data by zone.
    - Extracted thickness data using unique timestamps from the clean density data, ensuring consistency across datasets.

2. Exploratory Data Analysis

- **Visual Analysis**:
    - Conducted EDA using various plots:
        - Frequency distribution plots for all density values.
        - Scatter plots against datetime to visualize trends in the thickness dataset.
        - Correlation heatmaps to explore relationships between variables.
- **Initial Data Checks**:
    - Performed basic checks using functions like .head() and .describe() to understand the structure and summary statistics of the data.

3. Data processing

- **Data Cleaning**:
    - Applied standard data cleaning techniques, including:
        - Elimination of null and missing values.
        - Scaling of data using normalization techniques.
        - Removal of outliers using interquartile ranges (IQR) and specific criteria based on thickness data.
- **Outlier Treatment**:
    - Employed multiple outlier removal techniques, including:
        - IQR method for identifying and eliminating outliers.
        - Using thickness as a filter.

4. ML pipeline

- **Modeling Workflow**:
    - Implemented a comprehensive ML pipeline, experimenting with various models and included a few in the notebook. Rigorously hypertuned some, including Gradient Boosting and Random Forest Regressors.
- **Pipeline Steps**:
    - **Model Initialization**: Set up instances of selected ML models.
    - **Data Scaling**: Applied scaling to ensure model inputs are normalized.
    - **Data Splitting**: Divided data into training, validation, and test sets.
    - **Model Training and Evaluation**: Trained models on the training data and evaluated performance using metrics such as RMSE, MAPE, and R-squared on validation data.
    - **Hyperparameter Tuning**: Tuned models using validation data to optimize performance.
    - **Final Model Training**: Combined training and validation sets to retrain the model, followed by testing on the final test set.

Summary of the functions in the notebook

1. fetch_density_data

Fetches density data from Wonderware for a specific line within a time range.

2. processing

Processes raw data by removing null values and zeros, then aggregates the data into 10-minute intervals.

3. fetch_thickness_data

Fetches thickness data from Wonderware for a specific line within a time range.

4. get_thickness

Retrieves the thickness data for a given line.

5. plot_heatmap

Plots a heatmap for the given DataFrame to visualize the correlation.

6. plot_density

Plots the frequency distribution of the density data.

7. scale

Scales the data.

8. remove_outliers

Removes outliers from the DataFrame by filtering using IQR

9. filter_and_eliminate

Filters and eliminates rows from the density DataFrame based on the thickness data.

10. remove_outliers2

Removes outliers from the DataFrame using IQR

11. analyze_zones_and_cores

Rearrange and build new dataframes by mapping the zone values to core values in 2 ways per line.

12. plot_corr

Plots the correlation matrix for the given dataset.

13. pipeline

Executes the entire data science pipeline. Tests multiple ML models.

14. calculate_mape

Calculates the Mean Absolute Percentage Error (MAPE) between actual and forecasted values.

15. train_val_test_split

Splits the data into training, validation, and test sets based on provided ratios.

16. modeling

Performs model evaluation on the validation set and returns performance metrics.

17. model_tuning

Performs hyperparameter tuning using RandomizedSearchCV and GridSearchCV.

18. check_best_result

Checks the best result from a series of model evaluations.

19. check_result

Evaluates the model on the validation set and checks for performance metrics.

20. hypertune_model

Performs hyperparameter tuning using GridSearchCV or RandomizedSearchCV to find the best parameters for a given model.

21. plot_predictions

Plots the actual versus predicted values over time.

Extended Documentation

(Experimental Approaches and Additional Analyses not included in the notebook)

1. 5-Minute and 15-Minute Timeframes

The goal of experimenting with different timeframes (5-minute and 15-minute intervals) was to explore how the aggregation of data over shorter or longer periods might influence the insights drawn from the data and the performance of the models.

The raw data was resampled to create datasets averaging 5-minute and 15-minute intervals. To obtain results for these timeframes again, or experiment other intervals, the only change required is to update the number of minutes twice in the code: in the fetch_density_data() function and the processing() function.

2. Data manipulation techniques

Applied log, square root, exponential, reciprocal, and other mathematical transformations to the zone density values to explore nonlinear relationships and improve model performance. However, the correlation did not improve at all.

3. Normalizing Without StandardScaler

Tried to investigate the effects of normalization using methods other than StandardScaler. Defined a custom scaling function, but it was not required.

4. Other Exploratory Plots (e.g., Q-Q Plot)

Explored additional visualizations like the Q-Q (Quantile-Quantile) plot and several scatter plots to better analyze the data.

Also plotted scatter plots to visualize the predicted and actual IP Core values.

5. Variations during Outlier Removal

Experimented with different values and multiplicative factors for these bounds. Can be updated directly in the code if required.

6. General mapping and correlation

Constantly plotted the correlation matrices for both the lines while experimenting different techniques. However, in the final notebook, I only included selective results.

7. Other ML models

Explored the initial performance of way more models than those included in the notebook. Tried hypertuning other ML models as well, which didn't work well.

8. Final Core Density Analysis

Further analysis was conducted to check whether a board sample will pass the final core density test or not, using both the in-process and inline density values. This was all done locally in excel files and is not included in the code.