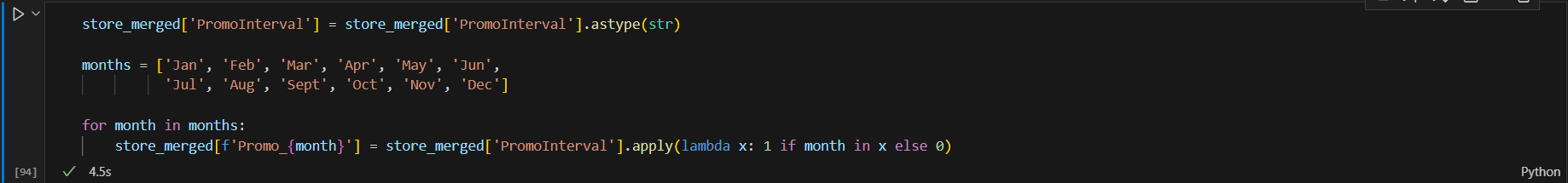
The goal of this feature engineering process is to prepare the *Rossmann Store Sales* dataset for machine learning and time series forecasting.  
This involves transforming categorical variables, imputing missing values logically and statistically, correcting data types, and scaling features for model readiness.



* The PromoInterval column originally contained month names (e.g., "Feb, May, Aug, Nov") indicating when recurring promotions occurred.
* The PromoInterval column originally contained month names (e.g., "Feb, May, Aug, Nov") indicating when recurring promotions occurred.
* It was first converted to a string to handle missing and mixed-type entries.
* Then, 12 binary columns (Promo\_Jan, Promo\_Feb, …, Promo\_Dec) were created to represent whether each month had an active promotion (1 = yes, 0 = no).
* The original text-based column was dropped after encoding.
* Purpose:
* Converting categorical month intervals into binary indicators improves model interpretability and compatibility, allowing algorithms to detect seasonal promotion effects numerically.
* It was first converted to a string to handle missing and mixed-type entries.
* Then, **12 binary columns (Promo\_Jan, Promo\_Feb, …, Promo\_Dec)** were created to represent whether each month had an active promotion (1 = yes, 0 = no).
* The original text-based column was dropped after encoding.

**Purpose:**

Converting categorical month intervals into binary indicators improves model interpretability and compatibility, allowing algorithms to detect seasonal promotion effects numerically.

A computer screen shot of text

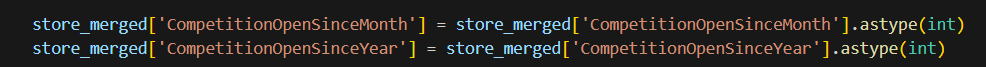
AI-generated content may be incorrect.

This section addresses missing or inconsistent values related to competition and promotions by combining **logical filling** and **statistical imputation**:

1. **CompetitionDistance Missing Values:**
   * Filled missing distances with a large placeholder value (200,000), representing **no nearby competitors**.
   * For these stores, corresponding competition start dates (CompetitionOpenSinceMonth and CompetitionOpenSinceYear) were set to 0, indicating non-applicability.
2. **Promo2 Campaign Logic:**
   * Stores not participating in secondary promotions (Promo2 = 0) had their related time references (Promo2SinceWeek and Promo2SinceYear) set to 0.
3. **Iterative Imputation for Remaining Gaps:**
   * Applied the **Iterative Imputer** (from sklearn.experimental) on remaining missing values in time-related columns.
   * This method predicts missing entries by modeling relationships among all features, ensuring statistically consistent imputations rather than arbitrary replacements.

**Purpose:**

* Ensures data completeness and logical consistency.
* Maintains realistic relationships between competition, promotions, and time variables.
* Reduces potential model bias from missing data while preserving feature correlations.



**Description:**

Converted the imputed month and year columns to integer types to restore valid time-based data formats.

**Purpose:**

Maintains proper data integrity for date-related operations, feature extraction, and visualization.

A screen shot of a computer

AI-generated content may be incorrect.

A black background with white text

AI-generated content may be incorrect.

Applied **Min-Max Scaling** to rescale all numerical features into the [0, 1] range.

**Purpose:**

* Standardizes the feature magnitudes to prevent dominance by large-scale values (e.g., CompetitionDistance).
* A black background with white text

  AI-generated content may be incorrect.Improves model performance and convergence for algorithms sensitive to input scale (e.g., neural networks, regression-based models).
* One hot encoder was used for the categorical columns.

**Time series analysis:**

1. A random store was chosen for each store type and visualized to test stationarity.
2. Rolling mean and standard deviation were made for the dataset with window = 7.
3. The Dickey-Fuller test was performed with its results proving that the data is suitable for time series machine learning models as it is stationary.
4. The data was then plotted for seasonality and trends every week and showed clear signs of both features existing in the dataset.
5. Data was further plotted using the Autocorrelation Function and the Partial Autocorrelation Function to further prove that the data has seasonality and is affected by trends.