Starengts Time Series Analysis Application User Manual

Introduction

Welcome to the STARENGTS Time Series Analysis Application, a powerful tool designed to facilitate comprehensive analysis of time series data. Whether you're working in manufacturing, finance, healthcare, or energy, this application offers an intuitive interface to upload, filter, and analyze your data using advanced visualization and statistical tools.

Features Overview

- **File Upload**: Upload .xlsx or .csv files for analysis.
- **Filter Options**: Filter data by date range, start and end time, value column, and sampling interval.
- **Visualizations**: Generate time series plots, box plots, time series decomposition, control charts, clustering plots, histograms, pair plots, and correlation heatmaps.
- **Descriptive Statistics**: View detailed statistical summaries, including total active and inactive times.
- **User Authentication**: Secure login with username and password.
- **Real-time Clock**: Displays a live clock on the application.
- **User Annotations**: Allows adding user annotations to the plots.
- **Advanced Analytics**: Supports polynomial regression and forecasting using Prophet.
- **Improved Data Display**: Utilizes st-aggrid for better data handling and pagination.
- **Caching**: Optimizes performance by caching results.
- **Error Handling**: Provides informative error messages for common issues.

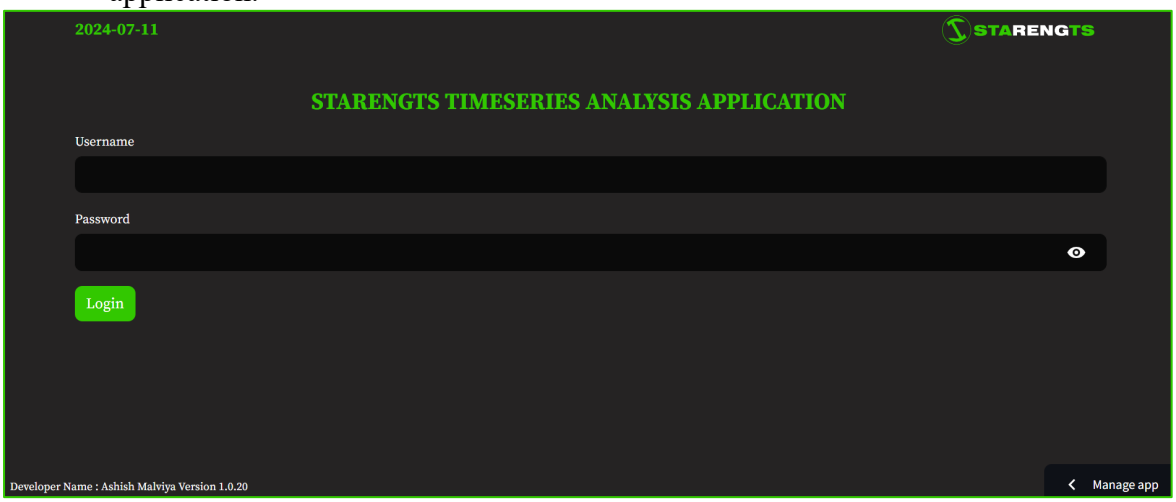
Getting Started

Access the Application

- Open the STARENGTS Time Series Analysis Application in your web browser. https://starengtstsapp.streamlit.app/

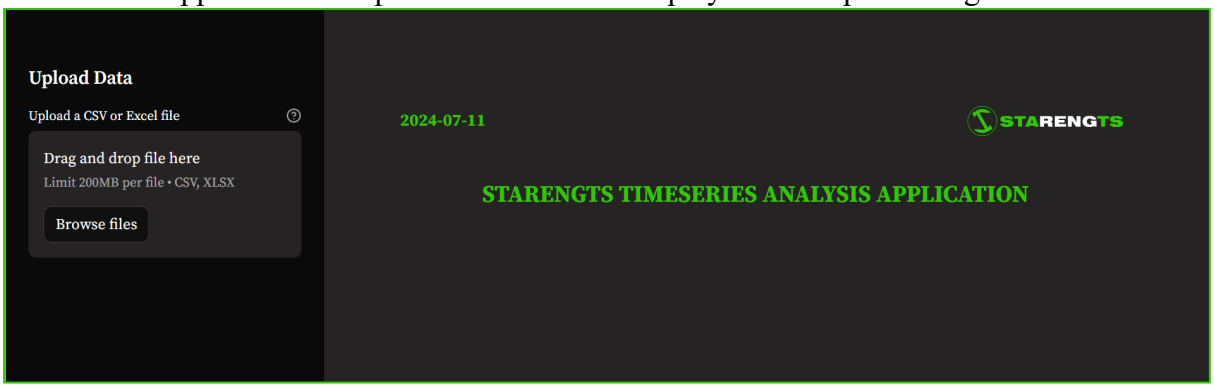
User Authentication

- Enter your username and password to log in.
- If your credentials are correct, you will be authenticated and granted access to the application.



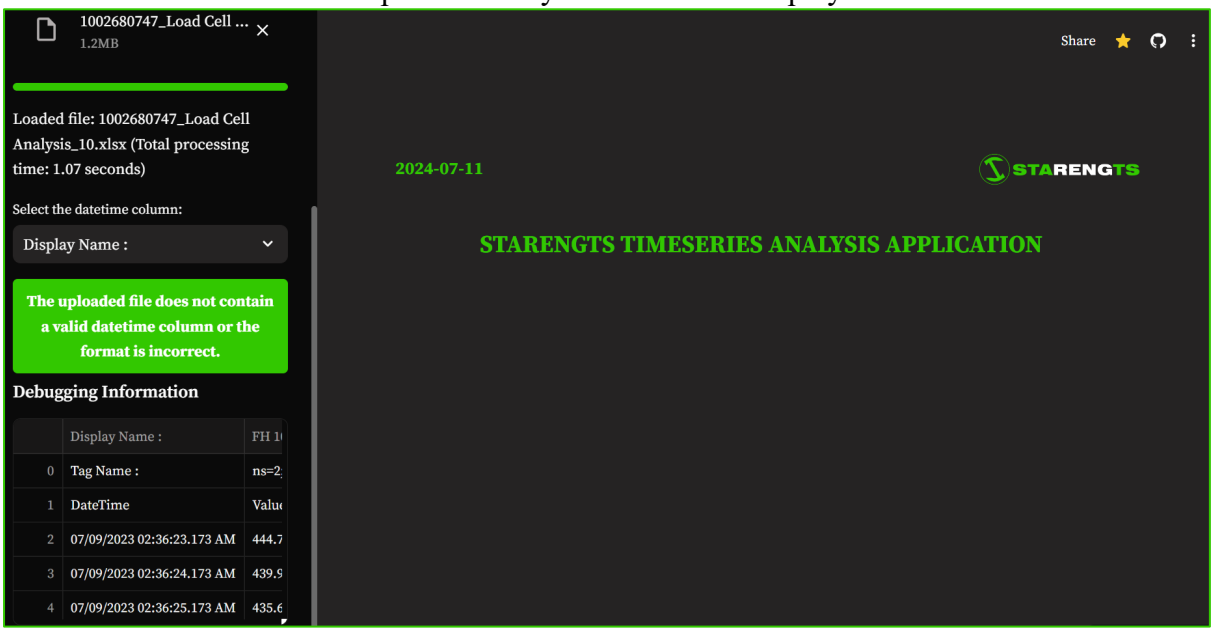
Upload Data

- Click on the "Upload File" button.
- Select a .xlsx or .csv file from your local system.
- The application will process the file and display the total processing time.



Load the file : Only Timeseries Data is allowed

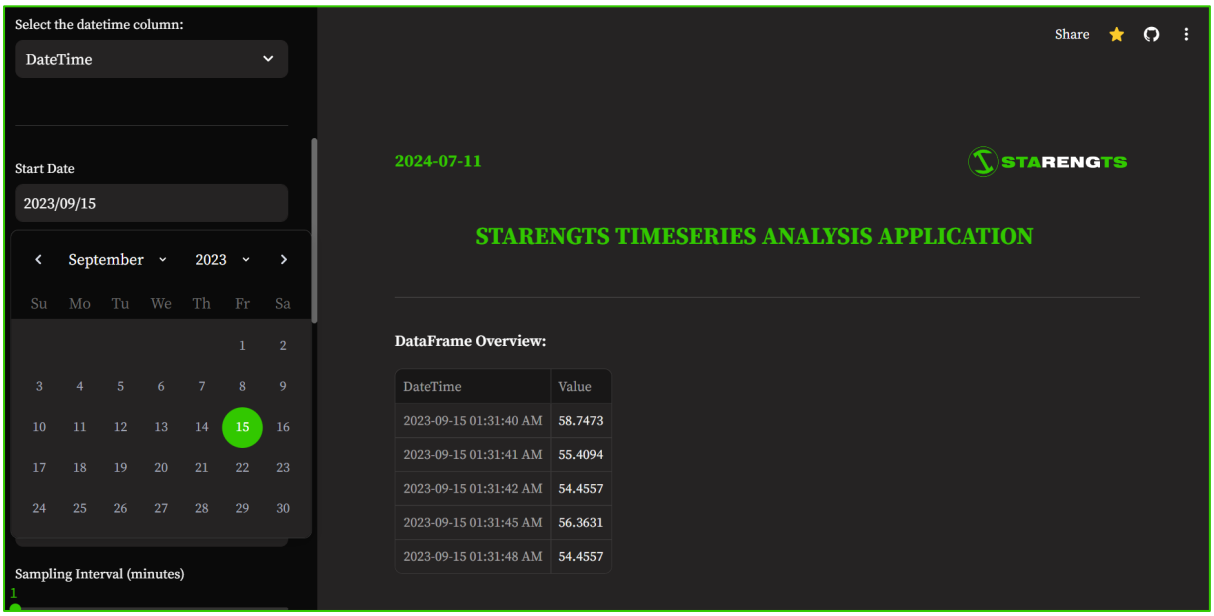
If no Timeseries Data in the uploaded file you will see the display error:



Using Filter Options

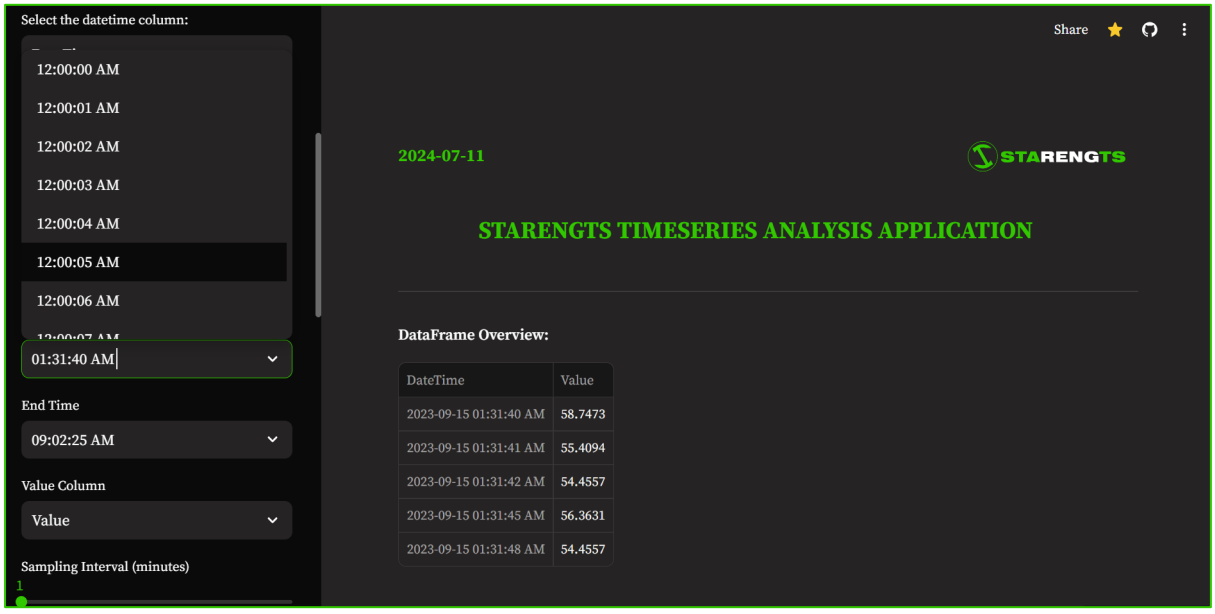
Date Range

- Use the "Date Range" picker to select the start and end dates for your analysis.



Start Time and End Time

- Use the "Start Time" and "End Time" dropdowns to specify the time range within the selected dates.



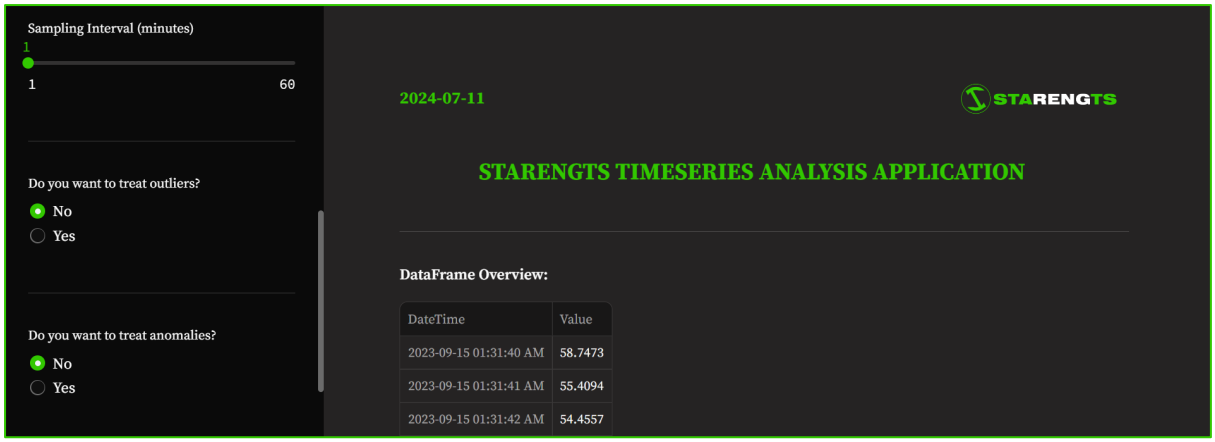
Value Column

- Select the column containing the values you want to analyze from the "Value Column" dropdown.

Sampling Interval

- Choose the desired sampling interval (in minutes) from the "Sampling Interval" dropdown.

Date-Time: You can also choose your date and time column.



Outlier Treatment: You can choose to do outliner treatment based on the data visualization output.

Anomaly Treatment: The visualization will show you the anomaly, you can also do treatment to remove the anomaly form the data set**.**

Data Overview:

DateTime	Value
2023-09-15 01:31:40 AM	58.7473
2023-09-15 01:31:41 AM	55.4094
2023-09-15 01:31:42 AM	54.4557
2023-09-15 01:31:45 AM	56.3631
2023-09-15 01:31:48 AM	54.4557

Data type info:

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 12337 entries, 2023-09-15 01:31:40.077000 to 2023-09-15 09:02:25.194000
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Value   12337 non-null  float64
dtypes: float64(1)
memory usage: 192.8 KB
```

Number of rows and columns in dataset:

```
(12337, 1)
```

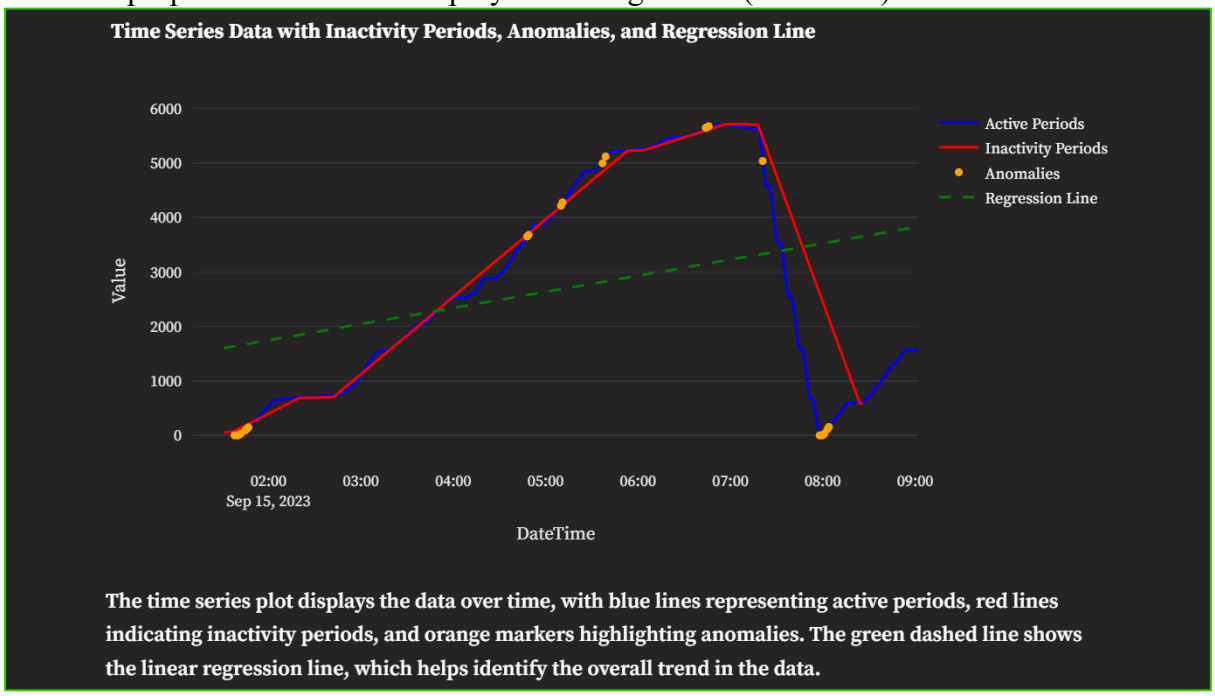
Total number of data points:

```
12337
```

Viewing Graphs and Plots

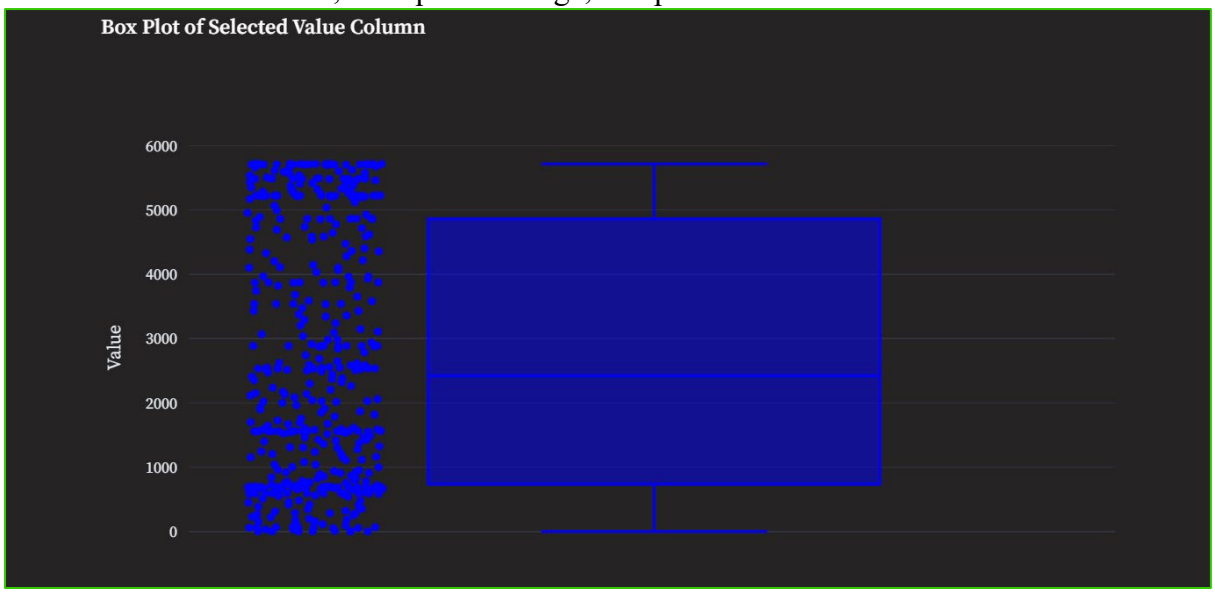
Time Series Plot

- Displays the data over time.
- Blue lines represent active periods.
- Red lines indicate inactivity periods.
- Orange markers highlight anomalies.
- A green dashed line shows the linear regression trend.
- A purple dotted line shows polynomial regression (if enabled)



Box Plot

- Visualizes the distribution of the selected data.
- Shows the median, interquartile range, and potential outliers.



Outlier and Anomaly treatment feedback:

Outliers detected by IQR method:

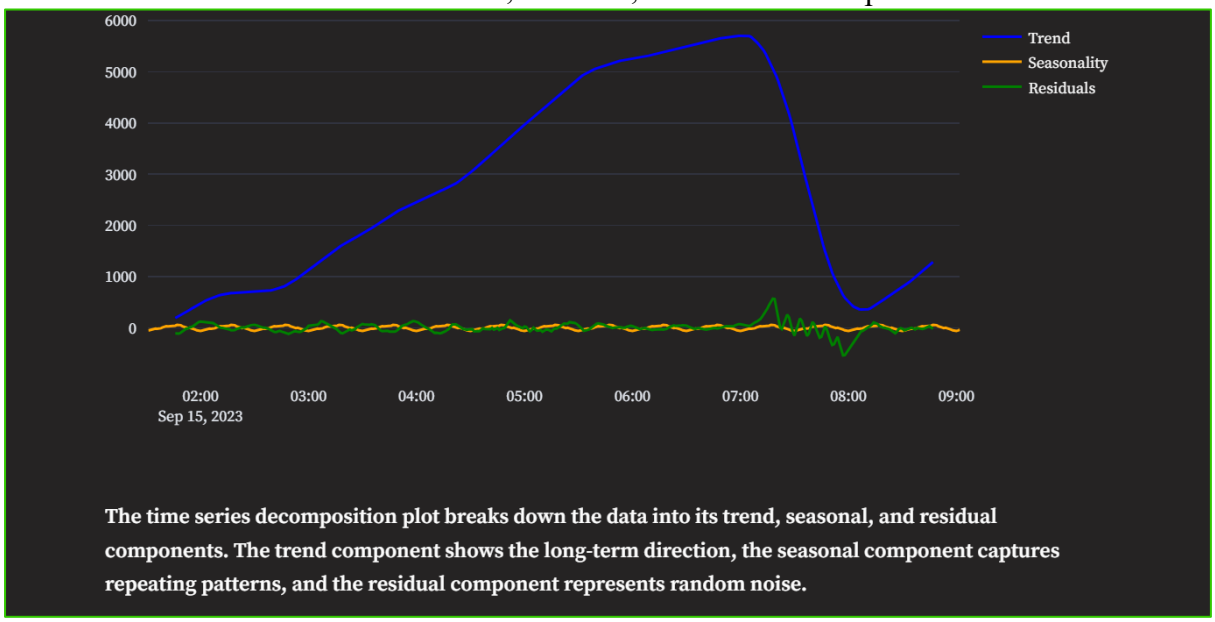
No outliers found in the dataset or you have done the treatment

Anomalies detected by Isolation Forest:

DateTime	Value	Anomaly
2023-09-15 01:38:00	0.8107	-1
2023-09-15 01:40:00	0.6518	-1
2023-09-15 01:41:00	11.4035	-1
2023-09-15 01:42:00	33.3156	-1
2023-09-15 01:45:00	94.335	-1

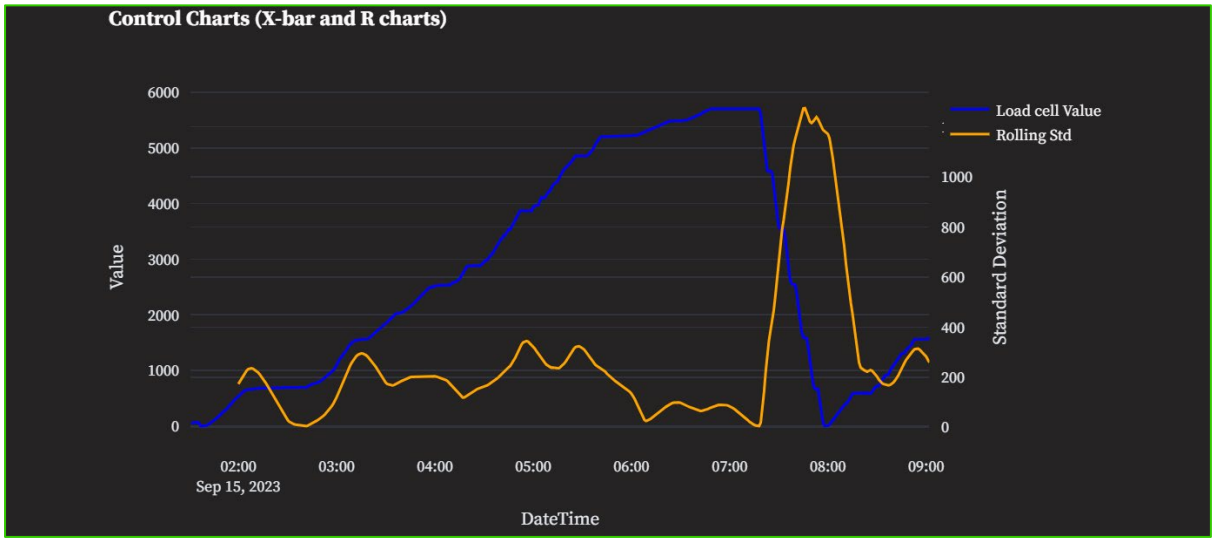
Time Series Decomposition Plot

- Breaks down the data into trend, seasonal, and residual components.



The time series decomposition plot breaks down the data into its trend, seasonal, and residual components. The trend component shows the long-term direction, the seasonal component captures repeating patterns, and the residual component represents random noise.

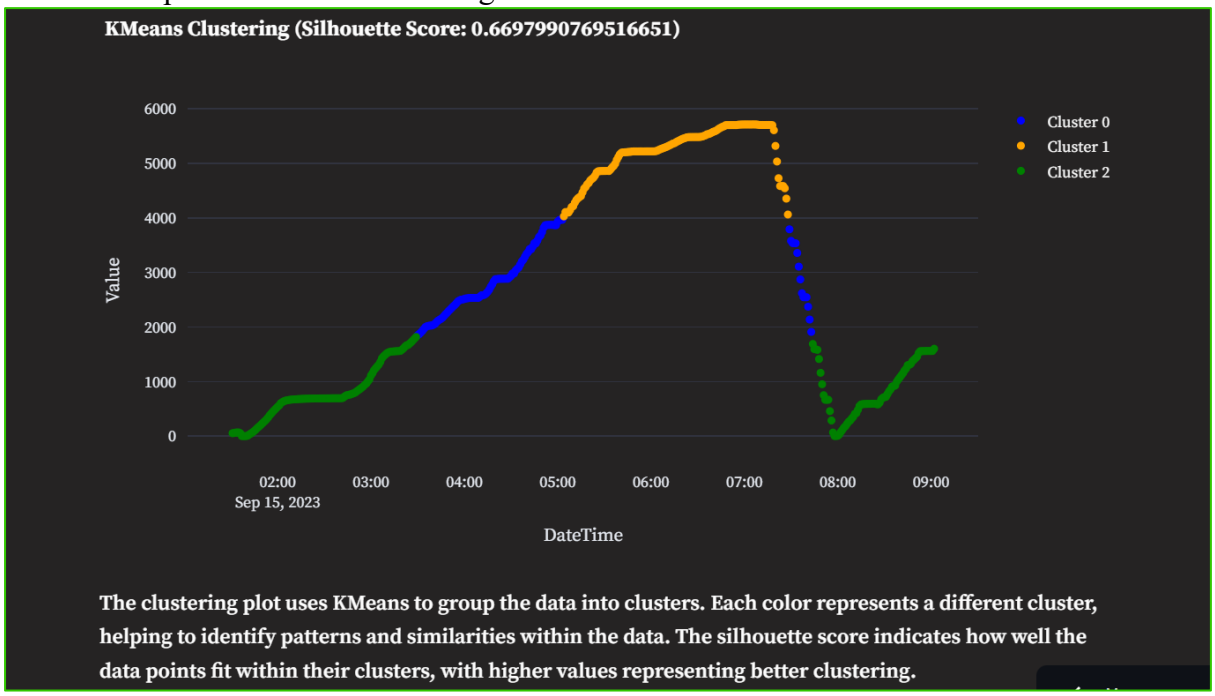
Control Chart



- Monitors process stability using X-bar and R charts.
- Displays load cell values and rolling standard deviation.

Clustering Plot

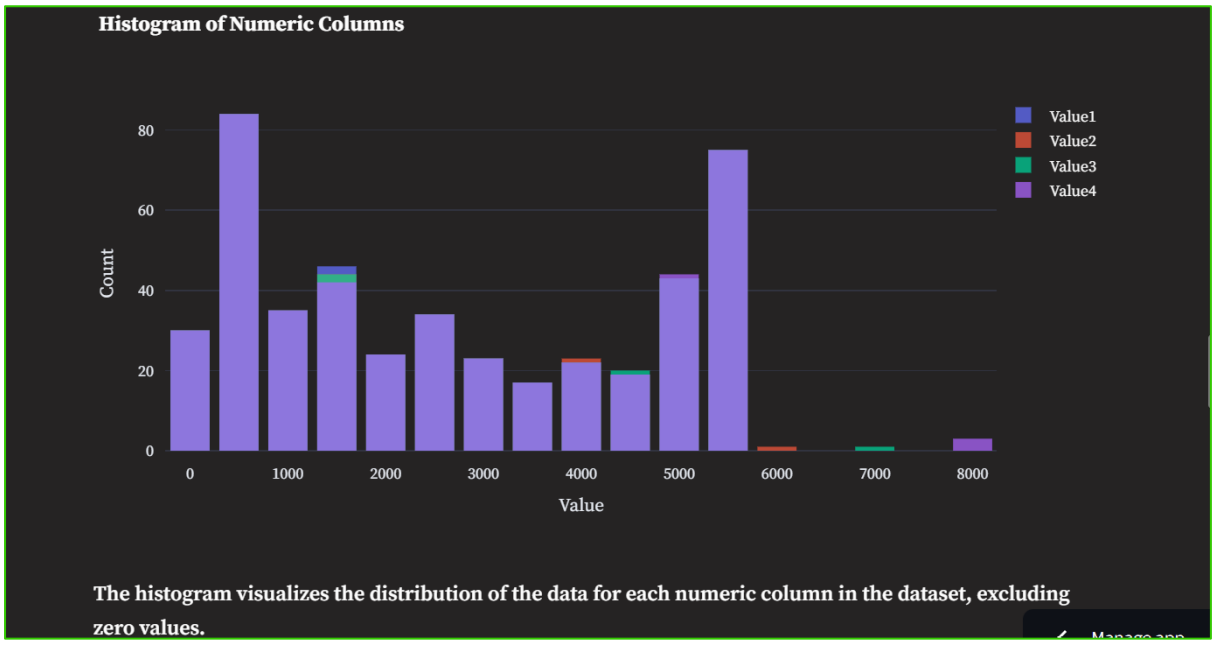
- Groups data into clusters using KMeans.



- Colors indicate different clusters, and the silhouette score evaluates clustering quality.

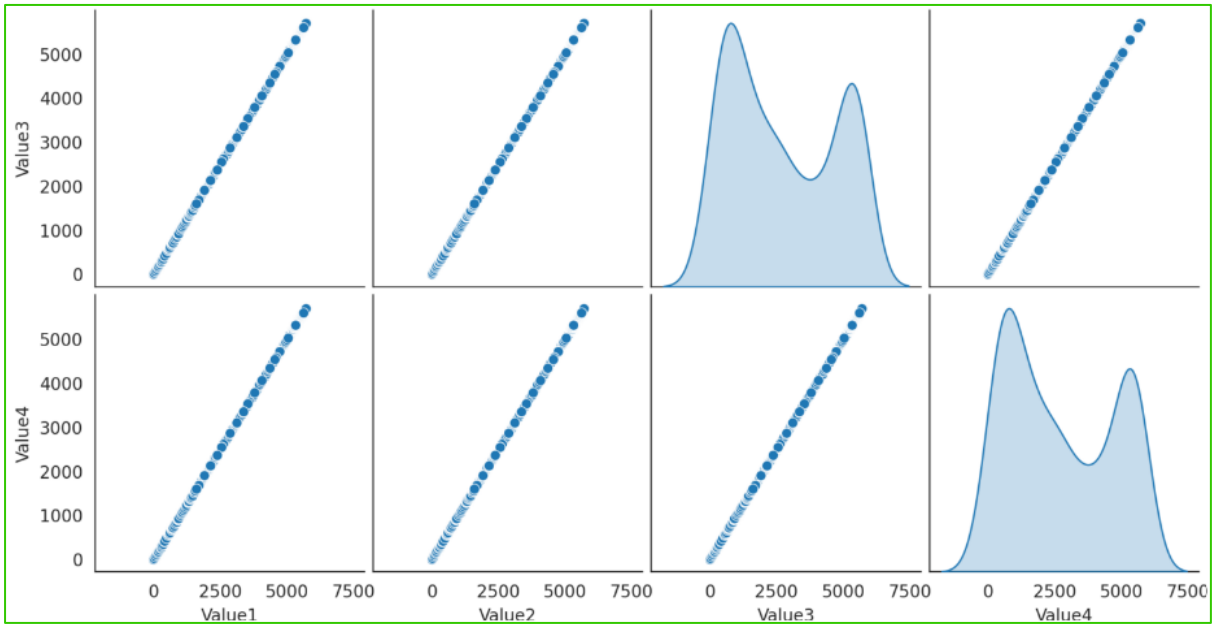
Histogram

- Visualizes the distribution of numeric columns in the dataset.



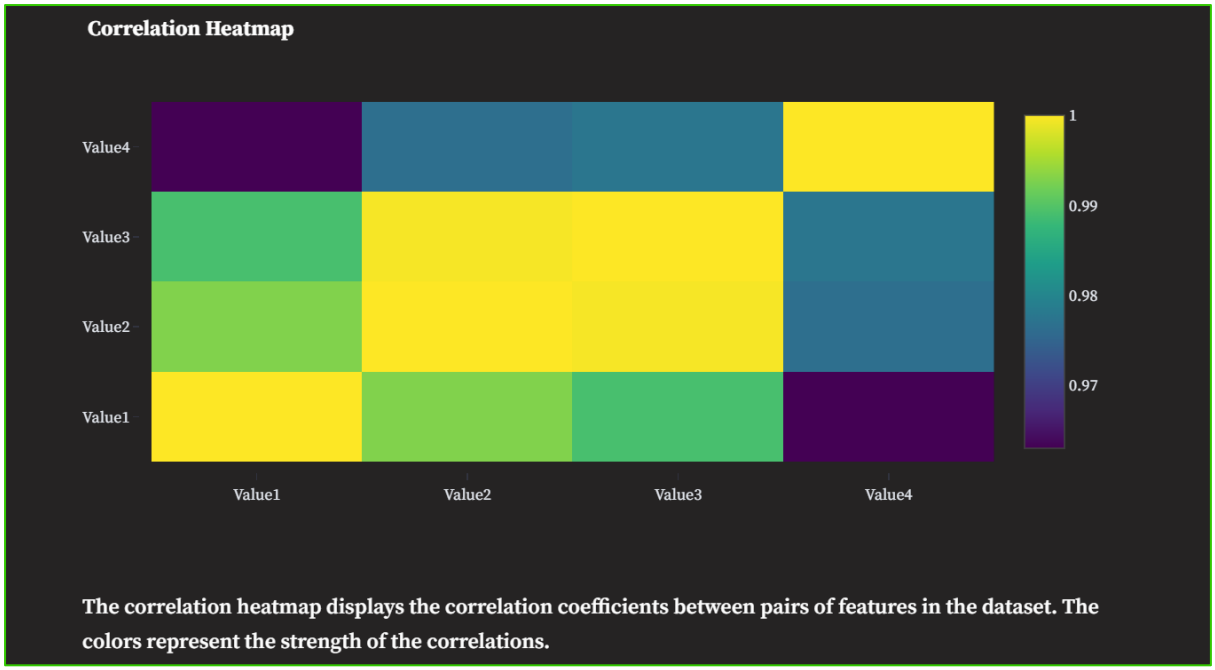
Pair Plot

- Displays pairwise relationships in the dataset, showing scatter plots for each pair of features and histograms for individual features.



Correlation Heatmap

- Displays the correlation coefficients between pairs of features in the dataset.



Descriptive Statistics

- View the detailed statistical summary, including:
  - Mean
  - Maximum
  - Minimum
  - Standard Deviation
  - Percentiles (25%, 50%, 75%)
  - Linear regression equation and R^2 value
  - Total active and inactive times

Advanced Analytics

Polynomial Regression

- Adjust the degree of polynomial regression using a slider.
- The polynomial regression line is displayed as a purple dotted line on the time series plot.

User Annotations

- Add text annotations to the plots.
- Enter the annotation text and specify the x and y values for placement.

Polynomial Regression

Polynomial regression is a type of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial. It is used when the data shows a non-linear relationship. The general form of a polynomial regression equation is:

$$y = b_0 + b_1 x + b_2 x^2 + \ldots + b_n x^n + \epsilon$$

where:

- y is the dependent variable.
- x is the independent variable.
- $b_0, b_1, b_2, \ldots, b_n$ are the coefficients of the polynomial.
- n is the degree of the polynomial.
- ϵ is the error term.

Components of the Plot

1. **Degree of Polynomial Regression Slider:**
   - This allows you to adjust the degree nnn of the polynomial regression. In the plot, it is set to 2, meaning a quadratic polynomial (degree 2) is used.
2. **Annotation Fields:**
   - These fields let you add custom annotations to specific points on the plot by specifying the text, xxx-value, and yyy-value.
3. **Active and Inactivity Periods:**
   - **Active Periods (Blue Line):** Periods where the values are actively changing.
   - **Inactivity Periods (Red Line):** Periods where the values are relatively constant or inactive.
4. **Anomalies (Orange Dots):**
   - Points that deviate significantly from the expected pattern, indicating potential anomalies.
5. **Regression Line (Green Dashed Line):**
   - The line of best fit for a linear regression, which models the relationship with a straight line.
6. **Polynomial Regression Line (Purple Dashed Line):**
   - The line of best fit for the polynomial regression of degree 2, which models the relationship with a quadratic curve.

Interpretation of the Plot

- The blue and red lines represent the data's active and inactive periods.
- The orange dots highlight anomalies in the data.

- The green dashed line represents a linear approximation of the relationship between the variables.
- The purple dashed line represents a quadratic approximation, which better fits the data if the relationship is non-linear.

Setting Values

To set the degree of polynomial regression:

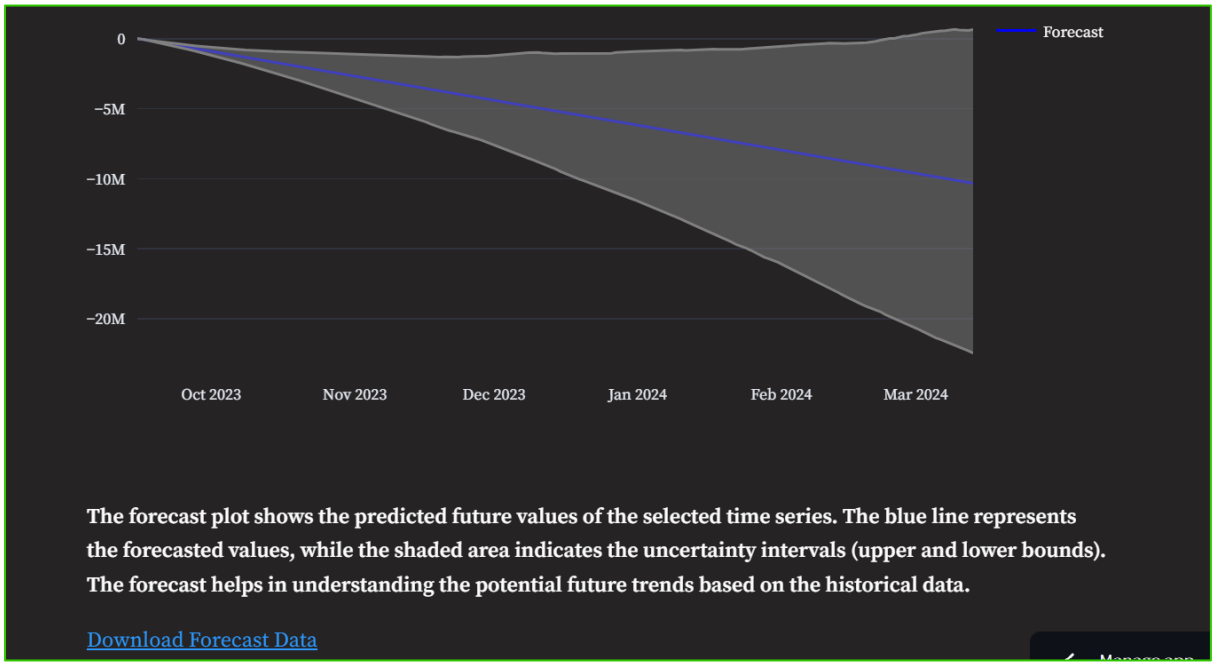
1. Use the slider labeled "Degree of Polynomial Regression" to select the desired polynomial degree.
2. The plot will update to show the polynomial regression line for the selected degree.

For annotations:

1. Enter the annotation text, xxx-value, and yyy-value in the respective fields.
2. Click the "Add Annotation" button to place the annotation on the plot.

Forecasting

- Use the Prophet library to forecast future values.
- Specify the forecasting period (in days).
- The forecast plot shows predicted future values with uncertainty intervals.
- Download the forecast data as a CSV file.



You can also download the manual from left slider.

Forecasting Feature

The forecasting feature in the STARENGTS Time Series Analysis Application leverages advanced time series forecasting techniques to predict future values based on historical data. This is particularly useful for planning, trend analysis, and decision-making in various domains such as manufacturing, finance, healthcare, and energy. Below is an in-depth look at the forecasting feature:

Tools and Libraries Used

- **Prophet**: Developed by Facebook, Prophet is a robust forecasting library designed for handling time series data with daily seasonality. It is highly effective for making forecasts in business applications and other domains that require accurate time series predictions.

How the Forecasting Feature Works

1. **Data Preparation**:
    - **Data Upload and Processing**: Users upload their time series data in .xlsx or .csv format. The application preprocesses the data, ensuring that the 'Timestamp' column is correctly parsed as datetime values.
    - **Filtering**: Users can filter the data by date range, time, value column, and sampling interval to select the most relevant subset of data for forecasting.
2. **Model Fitting**:
    - **Prophet Model**: The application utilizes the Prophet model to fit the historical data. The data is transformed into a format compatible with Prophet, with the 'Timestamp' column renamed to 'ds' (date) and the value column renamed to 'y' (value).
    - **Daily Seasonality**: Prophet automatically detects and incorporates daily seasonality into the model, making it suitable for data with regular patterns and cycles.
3. **Making Predictions**:
    - **Future Data Frame**: The application generates a future data frame representing the desired forecasting period (e.g., 180 days). This data frame includes dates beyond the historical data range for which predictions are needed.
    - **Forecast Generation**: Prophet uses the fitted model to predict future values. The forecast includes the predicted values ('yhat') as well as uncertainty intervals ('yhat_lower' and 'yhat_upper').
4. **Visualization**:
    - **Forecast Plot**: The forecast results are visualized using Plotly. The blue line represents the predicted future values ('yhat'), while the shaded area indicates the uncertainty intervals. This visualization helps users understand the range within which future values are likely to fall.
    - **Download Option**: Users can download the forecast data as a CSV file for further analysis and reporting.

Understanding Plot Descriptions

- Each plot is accompanied by a description explaining its components and how to interpret them.

Error Messages

- The application provides user-friendly error messages for:
  - Invalid file types
  - Missing 'Timestamp' column
  - No data available for the selected date range

Performance Optimization

- The application uses caching to optimize performance and avoid unnecessary re-computation.

Troubleshooting

- If you encounter any issues, ensure that your data file:
  - Is in .xlsx or .csv format
  - Contains a 'Timestamp' column with valid datetime values

1. **Camera Icon**: Save the chart as a PNG file. This allows you to export and save the current view of the chart as an image file.
2. **Zoom Icon**: Zoom in on a specific area of the chart. Click and drag to select the area you want to zoom into.
3. **Pan Icon**: Pan the chart. Click and drag to move around the chart. This is useful for exploring different parts of the chart without changing the zoom level.
4. **Zoom In Icon**: Zoom in incrementally. Clicking this will gradually zoom into the center of the chart.
5. **Zoom Out Icon**: Zoom out incrementally. Clicking this will gradually zoom out from the center of the chart.
6. **Auto-Scale Icon**: Reset axes to their default range. Clicking this will automatically scale the chart to fit all the data within the view.
7. **Reset Axes Icon**: Reset the chart to its original state. Clicking this will reset any zoom or pan actions and bring the chart back to its initial state when first rendered.
8. **Show Closest Data on Hover Icon**: Enable or disable hover information. Clicking this will toggle the display of data point information when hovering over the chart.

Here's a step-by-step explanation of the formula used in the code:

1. **Rolling Window Calculation:**

Python Copy code: inactivity_mask=(resampled_df[value_column].rolling('10T').max() - resampled_df[value_column].rolling('10T').min()) <= 15

- resampled_df[value_column].rolling('10T'): This applies a rolling window of 10 minutes over the time series data in the specified value column.
- .max(): Calculates the maximum value within each 10-minute rolling window.
- .min(): Calculates the minimum value within each 10-minute rolling window.
- The difference max() - min() gives the range of values within each rolling window.
- <= 15: Compares the range to the threshold (15 units). If the range is less than or equal to 15, the window is considered inactive.

2. **Identifying Inactive Periods:**

- inactivity_mask is a boolean mask where True indicates an inactive period (i.e., the activity range within the window is below the threshold).

3. **Filtering Active and Inactive Data:**

Python Copy code:

```
active_df = resampled_df[~inactivity_mask]
inactive_df = resampled_df[inactivity_mask]
```

- active_df contains the data points where the rolling window activity is above the threshold.
- inactive_df contains the data points where the rolling window activity is below the threshold.

4. **Calculating Total Active and Inactive Time:**

Python Copy code:

```
active_time = active_df.shape[0] * sampling_interval
inactive_time = inactive_df.shape[0] * sampling_interval
```

- active_df.shape[0]: Number of data points in the active periods.
- inactive_df.shape[0]: Number of data points in the inactive periods.
- Multiplying by sampling_interval (in minutes) converts the count of data points to total time. For example, if the sampling interval is 1 minute, each data point represents 1 minute of time.

5. **Displaying Active and Inactive Time:** These values are then displayed in the descriptive statistics section to provide insights into the proportion of time the system was active or inactive within the selected date and time range.

Frequently Asked Questions (FAQs)

Q1: What file formats are supported?

- The application supports .xlsx and .csv file formats.

Q2: How do I filter data by a specific time range?

- Use the "Start Time" and "End Time" dropdowns to select the desired time range within the selected dates.

Q3: What do the colors in the time series plot represent?

- Blue lines represent active periods, red lines indicate inactivity periods, and the green dashed line shows the linear regression trend.

Q4: What should I do if I see an error message?

- Check the error message details and ensure your data file meets the required format and contains the necessary columns.

How We Built the Application

The STARENGTS Time Series Analysis Application was developed using a combination of modern Python libraries and tools to ensure a robust, efficient, and user-friendly experience. Below is an overview of the key components and methodologies used in the development process:

Streamlit

- **Streamlit** was chosen as the main framework for building the web application. It allows for rapid development and deployment of interactive web apps using only Python, which significantly speeds up the development process.
- Key features of Streamlit utilized in this application include:
  - **File Uploads**: Enabling users to upload .xlsx and .csv files.
  - **Widgets**: Providing interactive elements like date pickers, sliders, and dropdowns for user input.
  - **Caching**: Optimizing performance by caching data and computational results.
  - **Real-time Updates**: Implementing a live clock and real-time updates using JavaScript embedded in Streamlit.

Data Handling and Processing

- **Pandas**: Used for data manipulation and analysis. It provides powerful data structures like DataFrame and Series, which are essential for handling time series data.
- **Numpy**: Utilized for numerical operations, particularly for handling and transforming data arrays.

Machine Learning and Statistical Analysis

- **Scikit-learn**: Employed for machine learning tasks such as linear regression, KMeans clustering, and anomaly detection using Isolation Forest.
- **Prophet**: A library developed by Facebook used for time series forecasting. It is particularly effective for handling time series data with daily seasonality.
- **Statsmodels**: Used for time series decomposition, breaking down the data into trend, seasonal, and residual components.

Visualization

- **Plotly**: A graphing library that makes interactive, publication-quality graphs online. It is used to create time series plots, box plots, decomposition plots, control charts, clustering plots, and more.
- **Seaborn**: Built on top of Matplotlib, it provides a high-level interface for drawing attractive and informative statistical graphics, such as pair plots and correlation heatmaps.

User Interface Enhancements

- **Streamlit-Aggrid**: Used for displaying large datasets efficiently with features like pagination and sorting. This enhances the user experience by allowing smooth browsing of large data tables.
- **Custom CSS and JavaScript**: Implemented for additional styling and functionality, such as the live clock.

Logging and Error Handling

- **Logging**: Python's built-in logging module is used to keep track of application events and errors, which helps in monitoring the application's performance and troubleshooting issues.
- **Error Handling**: The application is designed to handle common errors gracefully, providing informative messages to guide the user.

Authentication

- **Basic User Authentication**: Implemented to ensure that only authorized users can access the application. This feature enhances the security of the application by requiring valid credentials.

Expected Benefits of the Application

The STARENGTS Time Series Analysis Application is designed to provide users with a comprehensive and user-friendly tool for analyzing time series data. Below are the key benefits and functionalities users can expect from this application:

Streamlined Data Analysis

- **Easy Data Upload and Processing**: Users can easily upload their .xlsx or .csv files, and the application will automatically process and display the data, saving time and effort.
- **Interactive Filtering**: With interactive date pickers, time selectors, and other filtering options, users can quickly narrow down the data to the most relevant subsets for their analysis.

Advanced Visualization Tools

- **Time Series Plot**: Visualize data trends over time with clear indicators of active and inactive periods, anomalies, and regression lines.

- **Box Plot**: Understand the distribution of data, identify outliers, and get insights into data variability.
- **Decomposition Plot**: Decompose time series data into trend, seasonal, and residual components for more in-depth analysis.
- **Control Charts**: Monitor process stability and identify variations using X-bar and R charts.
- **Clustering Plot**: Group data into clusters to identify patterns and similarities within the data.
- **Histograms and Pair Plots**: Get detailed views of data distributions and relationships between variables.
- **Correlation Heatmaps**: Identify correlations between different features in the dataset.

Robust Statistical Analysis

- **Descriptive Statistics**: Access detailed statistical summaries including mean, maximum, minimum, standard deviation, and percentiles.
- **Linear and Polynomial Regression**: Perform regression analysis to identify trends and relationships within the data.
- **Anomaly Detection**: Automatically detect and highlight anomalies in the data using advanced machine learning algorithms.
- **Time Series Forecasting**: Use the Prophet library to predict future values and trends in the data, aiding in planning and decision-making.

User-Friendly Interface

- **Real-time Clock**: Keep track of the current time within the application for better context.
- **Annotations**: Add custom annotations to plots for better data storytelling and presentations.
- **Improved Data Handling**: Utilize `st-aggrid` for efficient data display and interaction, even with large datasets.

Performance Optimization

- **Caching**: Optimize performance and reduce processing times by caching results.
- **Efficient Data Handling**: Handle large datasets smoothly with pagination and efficient data processing techniques.

Secure Access

- **User Authentication**: Ensure secure access to the application by requiring valid login credentials.

Error Handling and Support

- **Informative Error Messages**: Get clear and actionable error messages to quickly resolve common issues.

- **Troubleshooting and Support**: Access to detailed troubleshooting steps and contact information for further assistance.

Application Use Cases

The STARENGTS Time Series Analysis Application is versatile and can be used in various scenarios including:

- **Manufacturing**: Monitor and analyze production data to improve process stability and efficiency.
- **Finance**: Analyze financial time series data to identify trends, anomalies, and make future predictions.
- **Healthcare**: Track patient data over time to identify trends and anomalies in health metrics.
- **Energy**: Monitor and forecast energy consumption and production data for better resource management.

By providing these functionalities, the STARENGTS Time Series Analysis Application helps users make informed decisions based on comprehensive and accurate data analysis.

How to Infer from the Application

The STARENGTS Time Series Analysis Application provides various tools and visualizations to help users analyze their time series data. Here's how to interpret the results and make informed decisions based on the analysis:

Time Series Plot

- **Active and Inactive Periods**: Blue lines represent periods of activity, while red lines indicate inactivity. By examining these periods, users can identify patterns and trends in the data over time.
- **Anomalies**: Orange markers highlight anomalies detected in the data. These points indicate unusual or unexpected values that may require further investigation.
- **Regression Lines**: The green dashed line shows the linear regression trend, which helps identify the overall direction of the data. The purple dotted line represents polynomial regression, providing a more detailed view of the data trend.

Box Plot

- **Data Distribution**: The box plot visualizes the distribution of the selected data, showing the median, interquartile range, and potential outliers. This helps users understand the central tendency and variability of the data.
- **Outliers**: Points outside the whiskers of the box plot indicate potential outliers. These outliers can be investigated to understand any unusual data points.

Time Series Decomposition Plot

- **Trend**: The trend component shows the long-term direction of the data. Users can infer whether the data is generally increasing, decreasing, or remaining stable over time.
- **Seasonal**: The seasonal component captures repeating patterns or cycles in the data. Identifying seasonality can help users understand periodic fluctuations.
- **Residuals**: The residual component represents random noise or irregular patterns in the data. Analyzing residuals helps in understanding the variability not explained by trend or seasonality.

Control Chart

- **Process Stability**: The control chart monitors the stability of a process over time. The X-bar chart shows the mean of the process, while the R chart displays the range of the process variation. Any significant deviations from the control limits may indicate process instability.
- **Rolling Standard Deviation**: The rolling standard deviation helps in identifying periods of increased variability. Higher standard deviation values indicate greater fluctuations in the data.

Clustering Plot

- **Clusters**: The clustering plot groups data into clusters using KMeans. Each color represents a different cluster, helping users identify patterns and similarities within the data.
- **Silhouette Score**: The silhouette score evaluates the quality of clustering. A higher score indicates better-defined clusters. Users can use this information to determine the effectiveness of the clustering algorithm.

Descriptive Statistics

- **Statistical Summary**: The detailed statistical summary includes measures such as mean, maximum, minimum, standard deviation, and percentiles. These metrics provide insights into the central tendency, spread, and overall distribution of the data.
- **Linear Regression**: The linear regression equation and R^2 value help in understanding the relationship between variables. A higher R^2 value indicates a stronger correlation.

Forecasting

- **Future Predictions**: The forecasting plot shows predicted future values based on historical data. The blue line represents forecasted values, while the shaded area indicates uncertainty intervals. Users can use these predictions for planning and decision-making.
- **Uncertainty Intervals**: The shaded area around the forecast line represents the range within which future values are likely to fall. Wider intervals indicate greater uncertainty.

User Annotations

- **Annotations**: Users can add custom annotations to the plots to highlight specific points of interest or provide additional context. This is useful for presentations and reporting.

Histograms and Pair Plots

- **Histograms**: Visualize the distribution of individual numeric columns. This helps in understanding the frequency and spread of data values.
- **Pair Plots**: Show relationships between pairs of features. Scatter plots for each pair of features and histograms for individual features help in identifying correlations and interactions.

Correlation Heatmap

- **Correlations**: The correlation heatmap displays the correlation coefficients between pairs of features. Strong positive or negative correlations can indicate relationships between variables that are worth exploring further.

Contact and Support

- For further assistance, please contact [info@starengts.com].

==Thank you for visiting us: Mail us for more information: info@starengts.com==