**Some information on specific columns:**

CLNDR\_DT = revenue transaction date

CLNDR\_DT\_MONTHSEQ = sequential month number that corresponds with CLNDR\_DT

Auth\_Date = The date from which a project is considered new. Because we do 12 month forecast, we look at this date +12 months as the date frame from where a project is new. For example if we are doing a forecast for January 2023- December 2023, if a project has an auth date of January 2023, we would not have historical data from <December 2022 so any revenue generated by this project would be considered ‘new’ for 2023.

Auth\_Date\_Monthseq = sequential month number that corresponds with Auth\_Date

**Question:**

How would you approach predicting what NEW PROJECT revenue will be for the next 12 months at the CATEGORY level. As noted above there will be no history for a new project. However, there are histories of projects that were considered ‘new’ from a particular point in time. The attached dataset was raw export with the data not filtered or cleaned up in anyway. Please review the data to be able to provide some specific details in answering the questions below.

* How would you think creating a ‘new project’ model? What type of approaches would you try?

**I utilized time series models based on past data to create average expense estimates by category (a total of 7 categories). Each new project is assigned a category, and using historical project data, we provide an average estimate of where their values might fall. Due to time constraints, we have focused only on Area1, but a thorough approach for estimating new projects by their area would involve this analysis for all categories. The table below will be used as input for forecasting both the year 2023 (filtering out the existing data up to May 2024) and for predictions from June 2024 to July 2025. We used machine learning models, specifically RandomForest, to carry out this analysis.**

* How would you transform this dataset so that it could be used for the approach you chose above?

A systematic approach involving multiple steps. Here is a detailed explanation of what we did:

#### 1. Data Preprocessing

* **Initial Data Load**: We started by loading the raw dataset containing various columns including CLNDR\_DT, Revenue, CLNDR\_DT\_MONTHSEQ, Auth\_Date, Auth\_Date\_Monthseq, Category, and project.
* **Datetime Conversion**: We converted the CLNDR\_DT and Auth\_Date columns from string format to datetime format to facilitate time series analysis.

#### 2. Feature Engineering

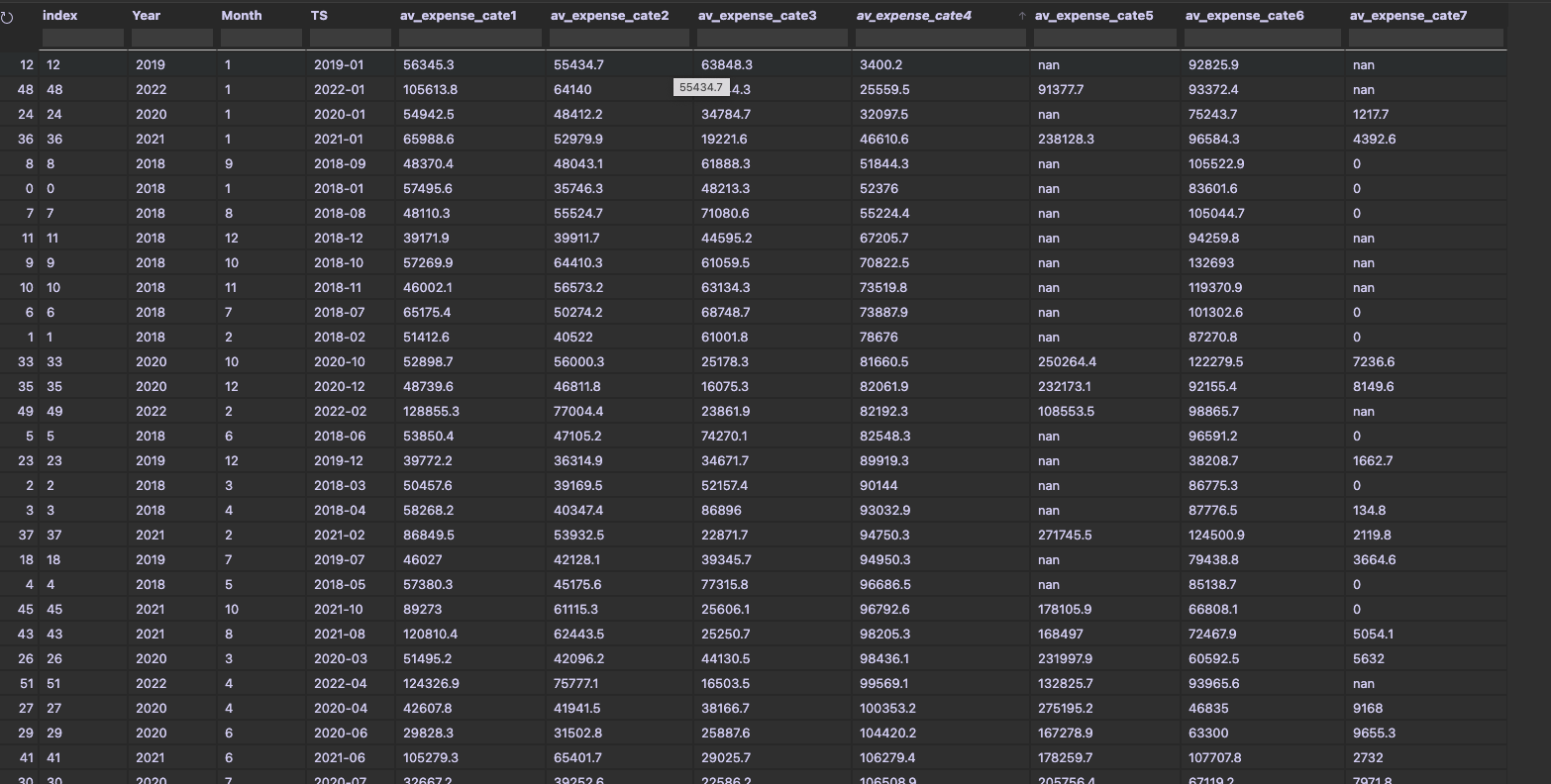
* **Extracting Time-Based Features**: We extracted the Year and Month from the CLNDR\_DT column to capture temporal patterns in the data. Additionally, we created a combined feature TS representing the year and month in the format 'YYYY-MM'.

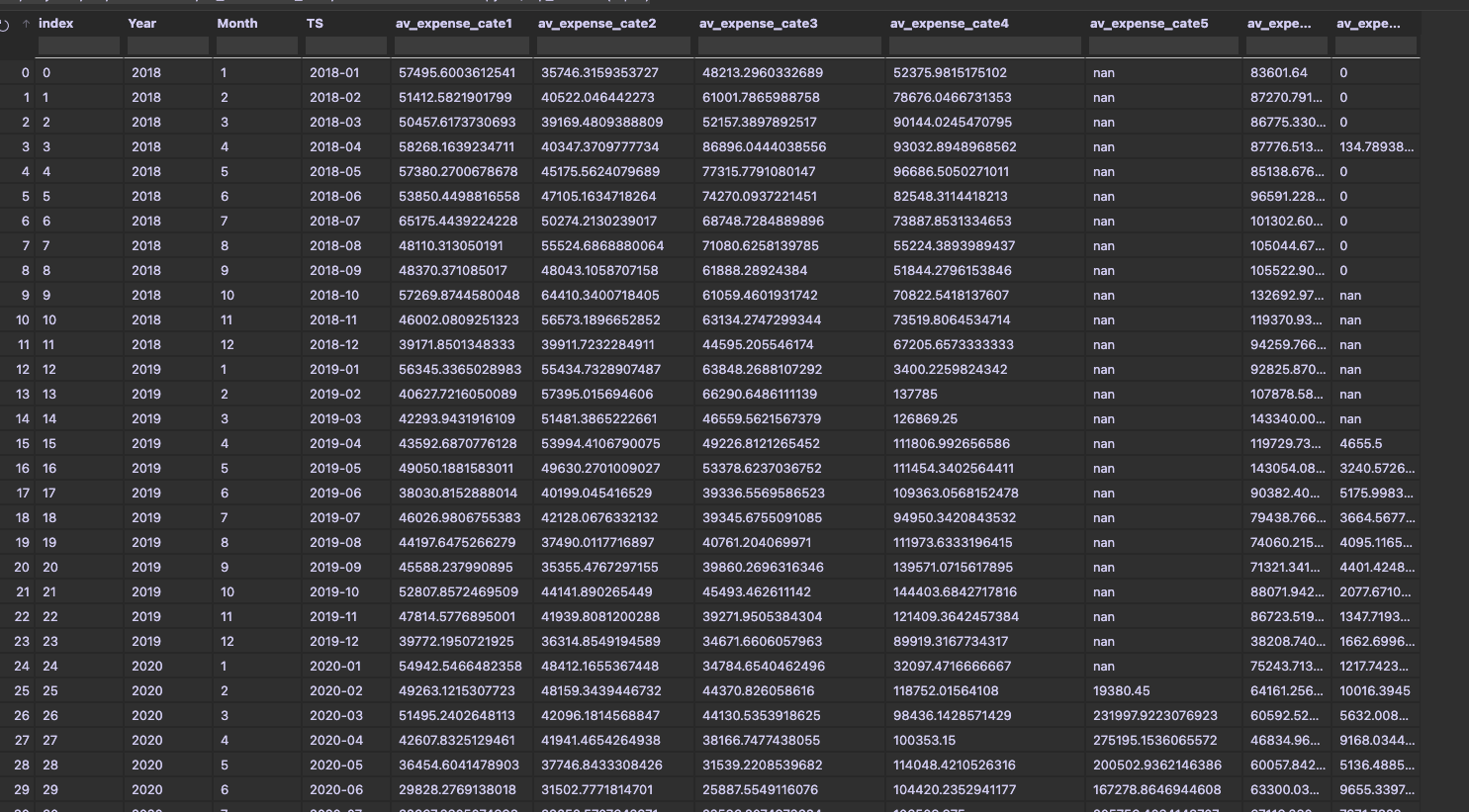
#### 3. Data Filtering and Grouping

* **Grouping by Category**: We grouped the data by Category to analyze expenses within each area. This was crucial since new projects would be assigned to one of these categories.
* **Calculating Average Expenses**: For each category, we calculated the average expenses (av\_expense\_cate1, av\_expense\_cate2, ..., av\_expense\_cate7) per month and year. This step provided a historical baseline for future predictions.

#### 4. Creating Training and Validation Sets

* **np20022\_forecast**: We created a dataset called np20022\_forecast which included data from January 2018 to December 2022. This dataset was used to understand historical trends and generate average expense estimates.
* **np\_forecast**: We created another dataset called np\_forecast which included data from January 2018 to May 2024. This dataset was used for both training and validation purposes, ensuring our model was well-calibrated before making future predictions.





* How would you filter data, would you create any new variables, etc? What would your test and training datasets look like? -🡪 **Please see the Jupyter Notebook**

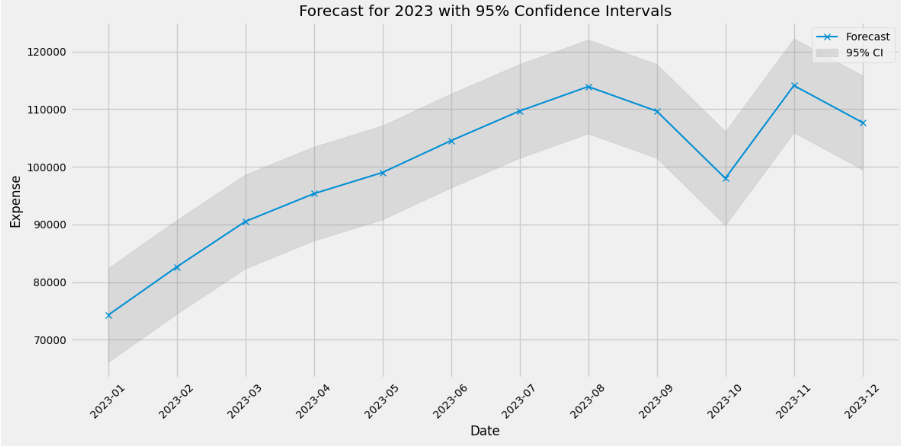
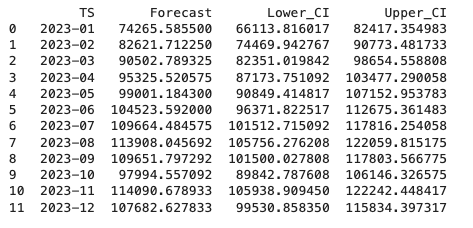
I took the following steps:

1. **Data Filtering**:
   * We filtered the dataset to include data from January 2018 to June 2023 for training, and from July 2023 to May 2024 for validation. This ensures the model is trained on sufficient historical data and validated before making future predictions.
2. **New Variables**:
   * We created new time-based features such as Year and Month to capture temporal patterns in the data.
   * Additionally, a combined feature TS representing the year and month in the format 'YYYY-MM' was created for easier time series analysis.
3. **Training and Test Datasets**:
   * The training dataset included data from January 2018 to June 2023.
   * The validation dataset included data from July 2023 to May 2024.
   * For future predictions, we extended the dataset to include the next 14 months (June 2024 to July 2025).
4. **Machine Learning Models**:
   * We used RandomForest, a machine learning model, to perform the forecasting. The model was trained using the training dataset and validated using the validation dataset. Predictions were then made for the future period from June 2024 to July 2025.

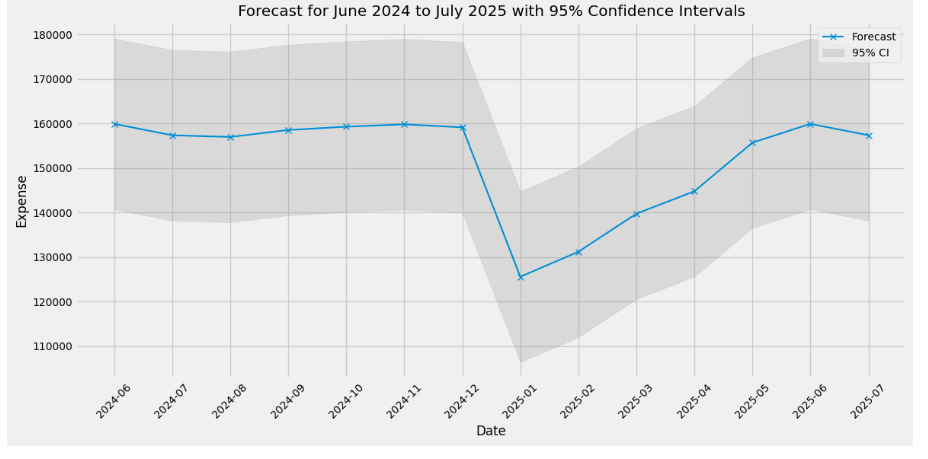
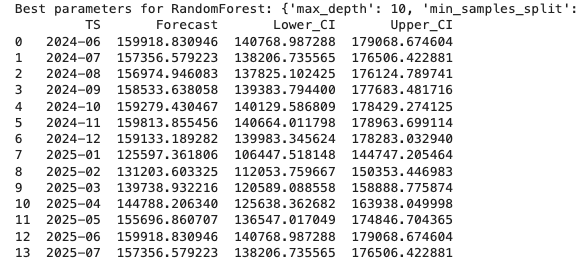
* Please provide examples of what these inputs into the model would be for (variables: dependent, independent, static, dynamic):

**Variables:**

1. **Dependent Variable**:
   * **av\_expense\_cate1**: This is the target variable we aim to forecast, representing the average expenses for Area1.
2. **Independent Variables**:
   * **Year**: Captures the yearly trend in the data.
   * **Month**: Captures the monthly seasonality in the data.
   * **TS**: A combined feature representing the year and month in the format 'YYYY-MM', useful for time series analysis.
3. **Static Variables**:
   * **Category**: The area or category to which the project belongs (in this case, Area1). This variable does not change over time and helps in grouping projects by their category.
4. **Dynamic Variables**:
   * **Year**: Changes over time and captures the annual trend.
   * **Month**: Changes monthly and captures seasonality effects.
   * Predicting from the beginning of 2023 (forecast for all new project revenue for full year (months 1-12).



* + Would your datasets look any different for predicting from July 2024 to July 2025?   
      
     **YES, revenues values changed over the time**



**Conclusion**In this analysis, I aimed to forecast expenses for new projects based on historical data. The steps we followed included:

1. **Data Preprocessing**: We filtered the dataset to include data from January 2018 to June 2023 for training, and from July 2023 to May 2024 for validation.
2. **Feature Engineering**: We created new time-based features such as Year, Month, and a combined feature TS representing the year and month in the format 'YYYY-MM'.
3. **Model Selection and Training**: We selected the RandomForest model for its robustness and applied hyperparameter tuning using GridSearchCV. The model was trained on the training dataset and validated using the validation dataset.
4. **Forecasting**: We generated forecasts for the period from June 2024 to July 2025, including confidence intervals to capture the uncertainty of predictions.
5. **Evaluation**: We evaluated the model's performance using metrics like MAD, MSE, AIC, BIC, and R², and validated it through a series of tests.

#### Limitations and Future Work

Due to time constraints, we limited our model selection to a few classic time series and machine learning models. In a comprehensive analysis, additional models such as ARIMA, ETS, Gradient Boosting Machines, and others should be considered to ensure robust forecasting. Deep learning models were not employed due to the limited number of data points (less than 100 periods), which is typically insufficient for effective deep learning model training.

#### Next Steps

1. **Deployment**: After finalizing the model selection, the next step would be to deploy the model in a production environment. This involves setting up a pipeline for continuous data integration and model retraining as new data becomes available.
2. **Monitoring and Validation**: Once deployed, it's crucial to monitor the model's performance to ensure it continues to provide accurate forecasts. Techniques such as A/B testing can be used to compare the performance of different models or scenarios.
3. **Performance Testing**: Simulating various scenarios and stress-testing the model can help identify potential weaknesses and areas for improvement. This involves testing the model under different conditions to see how well it generalizes to unseen data.

By following these steps, we can ensure that the forecasting model remains accurate and reliable, providing valuable insights for planning and decision-making in new projects.