<u>Some information on specific columns:</u>
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.

	index	Year	Month	TS	av_expense_cate1	av_expense_cate2	av_expense_cate3	av_expense_cate4	av_expense_cate5	av_expense_cate6	av_expense_cate7
12	12	2019	1	2019-01	56345.3	55434.7	63848.3	3400.2	nan	92825.9	nan
48	48	2022	1	2022-01	105613.8	64140	55434.7 [illegible]4.3	25559.5	91377.7	93372.4	nan
24	24	2020	1	2020-01	54942.5	48412.2	34784.7	32097.5	nan	75243.7	1217.7
36	36	2021	1	2021-01	65988.6	52979.9	19221.6	46610.6	238128.3	96584.3	4392.6
8	8	2018	9	2018-09	48370.4	48043.1	61888.3	51844.3	nan	105522.9	0
0	0	2018	1	2018-01	57495.6	35746.3	48213.3	52376	nan	83601.6	0
7	7	2018	8	2018-08	48110.3	55524.7	71080.6	55224.4	nan	105044.7	0
11	11	2018	12	2018-12	39171.9	39911.7	44595.2	67205.7	nan	94259.8	nan
9	9	2018	10	2018-10	57269.9	64410.3	61059.5	70822.5	nan	132693	nan
10	10	2018	11	2018-11	46002.1	56573.2	63134.3	73519.8	nan	119370.9	nan
6	6	2018	7	2018-07	65175.4	50274.2	68748.7	73887.9	nan	101302.6	0
1	1	2018	2	2018-02	51412.6	40522	61001.8	78676	nan	87270.8	0
33	33	2020	10	2020-10	52898.7	56000.3	25178.3	81660.5	250264.4	122279.5	7236.6
35	35	2020	12	2020-12	48739.6	46811.8	16075.3	82061.9	232173.1	92155.4	8149.6
49	49	2022	2	2022-02	128855.3	77004.4	23861.9	82192.3	108553.5	98865.7	nan
5	5	2018	6	2018-06	53850.4	47105.2	74270.1	82548.3	nan	96591.2	0
23	23	2019	12	2019-12	39772.2	36314.9	34671.7	89919.3	nan	38208.7	1662.7
2	2	2018	3	2018-03	50457.6	39169.5	52157.4	90144	nan	86775.3	0
3	3	2018	4	2018-04	58268.2	40347.4	86896	93032.9	nan	87776.5	134.8
37	37	2021	2	2021-02	86849.5	53932.5	22871.7	94750.3	271745.5	124500.9	2119.8
18	18	2019	7	2019-07	46027	42128.1	39345.7	94950.3	nan	79438.8	3664.6
4	4	2018	5	2018-05	57380.3	45175.6	77315.8	96686.5	nan	85138.7	0
45	45	2021	10	2021-10	89273	61115.3	25606.1	96792.6	178105.9	66808.1	0
43	43	2021	8	2021-08	120810.4	62443.5	25250.7	98205.3	168497	72467.9	5054.1
26	26	2020	3	2020-03	51495.2	42096.2	44130.5	98436.1	231997.9	60592.5	5632
51	51	2022	4	2022-04	124326.9	75777.1	16503.5	99569.1	132825.7	93965.6	nan
27	27	2020	4	2020-04	42607.8	41941.5	38166.7	100353.2	275195.2	46835	9168
29	29	2020	6	2020-06	29828.3	31502.8	25887.6	104420.2	167278.9	63300	9655.3
41	41	2021	6	2021-06	105279.3	65401.7	29025.7	106279.4	178259.7	107707.8	2732
30	30	2020	7	2020-07	32667.2	39252.6	22586.2	106508.9	205756.4	67119.2	7971.8

	index	Year	Month	TS	av_expense_cate1	av_expense_cate2	av_expense_cate3	av_expense_cate4	av_expense_cate5	av_expe...	av_expe...
0	0	2018	1	2018-01	57495.6003612541	35746.3159353727	48213.2960332689	52375.9815175102	nan	83601.64	0
1	1	2018	2	2018-02	51412.5821901799	40522.046442273	61001.7865988758	78676.0466731353	nan	87270.791...	0
2	2	2018	3	2018-03	50457.6173730693	39169.4809388809	52157.3897892517	90144.0245470795	nan	86775.330...	0
3	3	2018	4	2018-04	58268.1639234711	40347.3709777734	86896.0444038556	93032.8948968562	nan	87776.513...	134.78938...
4	4	2018	5	2018-05	57380.2700678678	45175.5624079689	77315.7791080147	96686.5050271011	nan	85138.676...	0
5	5	2018	6	2018-06	53850.4498816558	47105.1634718264	74270.0937221451	82548.3114418213	nan	96591.228...	0
6	6	2018	7	2018-07	65175.4439224228	50274.2130239017	68748.7284889896	73887.8531334653	nan	101302.60...	0
7	7	2018	8	2018-08	48110.313050191	55524.6868880064	71080.6258139785	55224.3893989437	nan	105044.67...	0
8	8	2018	9	2018-09	48370.371085017	48043.1058707158	61888.28924384	51844.2796153846	nan	105522.90...	0
9	9	2018	10	2018-10	57269.8744580048	64410.3400718405	61059.4601931742	70822.5418137607	nan	132692.97...	nan
10	10	2018	11	2018-11	46002.0809251323	56573.1896652852	63134.2747299344	73519.8064534714	nan	119370.93...	nan
11	11	2018	12	2018-12	39171.8501348333	39911.7232284911	44595.205546174	67205.6573333333	nan	94259.766...	nan
12	12	2019	1	2019-01	56345.3365028983	55434.7328907487	63848.2688107292	3400.2259824342	nan	92825.870...	nan
13	13	2019	2	2019-02	40627.7216050089	57395.015694606	66290.6486111139	137785	nan	107878.58...	nan
14	14	2019	3	2019-03	42293.9431916109	51481.3865222661	46559.5621567379	126869.25	nan	143340.00...	nan
15	15	2019	4	2019-04	43592.6870776128	53994.4106790075	49226.8121265452	111806.992656586	nan	119729.73...	4655.5
16	16	2019	5	2019-05	49050.1881583011	49630.2701009027	53378.6237036752	111454.3402564411	nan	143054.08...	3240.5726...
17	17	2019	6	2019-06	38030.8152888014	40199.045416529	39336.5569586523	109363.0568152478	nan	90382.40...	5175.9983...
18	18	2019	7	2019-07	46026.9806755383	42128.0676332132	39345.6755091085	94950.3420843532	nan	79438.766...	3664.5677...
19	19	2019	8	2019-08	44197.6475266279	37490.0117716897	40761.204069971	111973.6333196415	nan	74060.215...	4095.1165...
20	20	2019	9	2019-09	45588.237990895	35355.4767297155	39860.2696316346	139571.0715617895	nan	71321.341...	4401.4248...
21	21	2019	10	2019-10	52807.8572469509	44141.890265449	45493.462611142	144403.6842717816	nan	88071.942...	2077.6710...
22	22	2019	11	2019-11	47814.5776895001	41939.8081200288	39271.9505384304	121409.3642457384	nan	86723.519...	1347.7193...
23	23	2019	12	2019-12	39772.1950721925	36314.8549194589	34671.6606057963	89919.3167734317	nan	38208.740...	1662.6996...
24	24	2020	1	2020-01	54942.5466482358	48412.1655367448	34784.6540462496	32097.4716666667	nan	75243.713...	1217.7423...
25	25	2020	2	2020-02	49263.1215307723	48159.3439446732	44370.826058616	118752.01564108	19380.45	64161.256...	10016.3945
26	26	2020	3	2020-03	51495.2402648113	42096.1814568847	44130.5353918625	98436.1428571429	231997.9223076923	60592.52...	5632.008...
27	27	2020	4	2020-04	42607.8325129461	41941.4654264938	38166.7477438055	100353.15	275195.1536065572	46834.96...	9168.0344...
28	28	2020	5	2020-05	36454.6041478903	37746.8433308426	31539.2208539682	114048.4210526316	200502.9362146386	60057.842...	5136.4885...
29	29	2020	6	2020-06	29828.2769138018	31502.7771814701	25887.5549116076	104420.2352941177	167278.8646944608	63300.03...	9655.3397...

- 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.

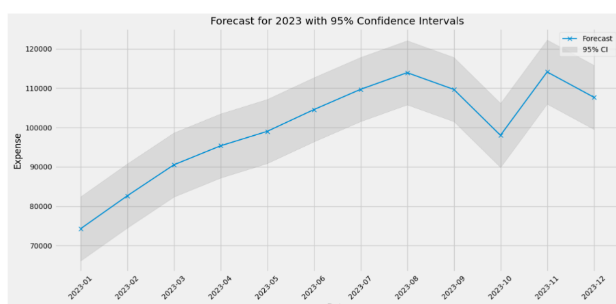
- Please provide examples of what these inputs into the model would be for (variables: dependent, independent, static, dynamic):

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).
-

	TS	Forecast	Lower_CI	Upper_CI
0	2023-01	74265.585500	66113.816017	82417.354983
1	2023-02	82621.712250	74469.942767	90773.481733
2	2023-03	90502.789325	82351.019842	98654.558808
3	2023-04	95325.520575	87173.751092	103477.290058
4	2023-05	99001.184300	90849.414817	107152.953783
5	2023-06	104523.592000	96371.822517	112675.361483
6	2023-07	109664.484575	101512.715092	117816.254058
7	2023-08	113908.045692	105756.276208	122059.815175
8	2023-09	109651.797292	101500.027808	117803.566775
9	2023-10	97994.557092	89842.787608	106146.326575
10	2023-11	114090.678933	105938.909450	122242.448417
11	2023-12	107682.627833	99530.858350	115834.397317

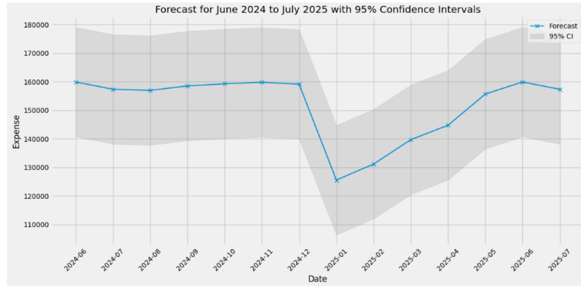


- Would your datasets look any different for predicting from July 2024 to July 2025?

YES, revenues values changed over the time

```
Best parameters for RandomForest: {'max_depth': 10, 'min_samples_split':
```

	TS	Forecast	Lower_CI	Upper_CI
0	2024-06	159918.830946	140768.987288	179068.674604
1	2024-07	157356.579223	138206.735565	176506.422881
2	2024-08	156974.946083	137825.102425	176124.789741
3	2024-09	158533.638058	139383.794400	177683.481716
4	2024-10	159279.430467	140129.586809	178429.274125
5	2024-11	159813.855456	140664.011798	178963.699114
6	2024-12	159133.189282	139983.345624	178283.032940
7	2025-01	125597.361806	106447.518148	144747.205464
8	2025-02	131203.603325	112053.759667	150353.446983
9	2025-03	139738.932216	120589.088558	158888.775874
10	2025-04	144788.206340	125638.362682	163938.049998
11	2025-05	155696.860707	136547.017049	174846.704365
12	2025-06	159918.830946	140768.987288	179068.674604
13	2025-07	157356.579223	138206.735565	176506.422881



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 R^2, 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.