The Churn Project

Task 3: Feature Engineering & Modelling

Well done for your analysis on the influence of price sensitivity relative to churn!

Sr. Data Scientist reviewed your work with the AD and Sr. Data Scientist has come up with an idea to enrich the dataset when trying to predict churn:

- *"I think that the **difference between off-peak prices in December and January the preceding year** could be a significant feature when predicting churn"*

As the Data Scientist on the team, you need to investigate this question. So, in this task you'll be responsible for completing feature engineering for the dataset.

What is feature engineering?

Feature engineering refers to:

- **Addition**
- **Deletion**
- **Combination**
- **Mutation**

of your data set to improve machine learning model training, leading to better performance and greater accuracy.

In context of this task, feature engineering refers to the engineering of the price and client data to create new columns that will help us to predict churn more accurately.

Effective feature engineering is based on sound knowledge of the business problem and the available data sources.

Creating the new features

The Sr. Data Scientist has done some further cleaning of the data and provided you with a new CSV file to complete our work from named "clean_data_after_eda.csv". Be sure to use this data for your work on this task.

Here's what you need to think about before you submit your work.

Your task is to create new features for your analysis and upload your completed python file.

The Churn Project

As before, a good way to quickly learn how to effectively feature engineer is to build a framework to follow. Below is an example of how you could attempt this task:

First - can we remove any of the columns in the datasets?

- There will almost always be columns in a dataset that can be removed, perhaps because they are not relevant to the analysis, or they only have 1 unique value.

Second - can we expand the datasets and use existing columns to create new features?

- For example, if you have "date" columns, in their raw form they are not so useful. But if you were to extract month, day of month, day of year and year into individual columns, these could be more useful.

Third - can we combine some columns together to create "better" columns?

- How do we *define* a "better" column and how do we *know which* columns to combine?
    - We're trying to accurately predict churn - so a "better" column could be a column that improves the accuracy of the model.
    - And which columns to combine? This can sometimes be a matter of experimenting until you find something useful, or you may notice that 2 columns share very similar information so you want to combine them.

Finally - can we combine these datasets and if so, how?

- To combine datasets, you need a column that features in both datasets that share the same values to join them on.

At this stage, your data could look vastly different, or may have just some subtle differences to how it was before.

You will be done with this task when you're happy with the new set of features that you've created and you think you're ready to build a predictive model to see which of these features are useful for predicting churn. Upload your python file and move onto the example answer.