Task 3: Feature Engineering & Modelling

Task Overview

What you'll learn

- How feature engineering can be used to test hypotheses
- How to build features to analyse the data for PowerCo

What you'll do

- Use Python to build a new feature for your analysis

Now it’s time for feature engineering

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

Estelle reviewed your work with the AD and Estelle 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.

Before we start on this task, let’s explain what feature engineering is…

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

Estelle 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.

To help you understand her idea about the new feature to create, Estelle has also provided a starter notebook which will help you to create the feature she described. This notebook is called “feature_engineering.ipynb”. Use this as a template to start your feature engineering.

We’ll show you some tips on the next step before you start your task, but make sure you have the relevant files downloaded before moving on!

- Download the new CSV data and Jupyter notebook
- Run the cells in the Jupyter notebook to create Estelle’s suggested feature
- We'll continue working in this Jupyter notebook to create some more columns, it's time to get creative!

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. We'll show you an example answer on the next step, but we encourage you to give it a go first! Below are some tips on how to get started.

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.

Explanation:

Set up:

- This task is focused on feature engineering, Estelle has provided a CSV dataset for you to use as a base for this task.
- You should download the notebook and CSV and start by running the cells within the notebook.
- Estelle has also provided some insight into a feature that would be interesting to add to the data. By running the cells in the notebook, this will create the feature that she described for you and provide a foundation for you to begin your own feature engineering.

Here is some context around the additional features that have been engineered in the notebook, to help you in the future:

- Firstly we have the average price changes across periods. This is a measure of the average price change by company between peak, mid-peak and off peak periods.
- We then take this idea one step further by creating another similar feature but instead of looking at the average price difference, we look at the maximum price difference across periods and months. This gives another way to look at the price changes across months.
- The reason why these 2 features could be useful is because they are another way of representing the variance of prices throughout the year. Imagine, if your utilities bill massively increased over winter, as a consumer you'd be annoyed and want to find a better deal!
- After this we continue feature engineering with some more concepts, including transformation of columns.
- To make predictions with a statistical or machine learning algorithm, all of the data must be converted to numeric data types.
- Therefore, we convert date into months and remove the raw date column, as we cannot use it in its original form.
- We also convert boolean columns into binary values.

- And we convert categorical columns into dummy variables. A dummy variable is a binary flag that indicates when a row matches the value from the categorical column that it was created from.
- As we saw during exploratory data analysis, the distribution of some columns was skewed. This is important to identify because when modeling data for prediction, based on the technique or algorithm that we use, there are sometimes assumptions within the data that we should follow.
    - One common assumption is that the columns within the data are normally distributed. Hence, if we find that columns are not normally distributed, we should treat these columns to try and transform them into a distribution that is more normal.
- Therefore, the next thing we do is transform some columns to have a closer to normal distribution. We do this using the logarithm function. As you can see from the visualisations, the newly transformed columns are much closer to a normal distribution than what they were earlier.
- Finally, we plot correlations of all the columns to see if we can identify any columns to remove. Columns that have very high correlations indicate an area to look out for. In this case, you may want to remove one of the columns, since they are likely both holding very similar information.