Mobile Phone Decision Tree Classifier-Documentation.

This document contains guidance on **Scripting a Python Model into PowerBI**, using the mobile phone data; a decision tree model to predict the mobile phone price band is developed inside PowerBI using Python script, with the resulting model used to populate PowerBI visual reports. This example can be reused for any similar predictive scenario that you want to embed in PowerBI to share reports in the business.

TIP : Power BI is highly compatible with Pandas. If you cannot find an array or series that you have processed, try combining it with other columns and turn it into a Data Frame. You can also create a one column data frame, convert a series or array to a data frame.

- https://pandas.pydata.org/docs/reference/api/pandas.Series.to_frame.html

- https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

Setup for running Python in PowerBI:

- Running python scripts in PowerBI requires some initial setup – this link includes the path configuration and a simple test python script as a data source into the powerbi workbook

- https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-python-scripts

- Ensure that required python libraries [numpy, pandas, scikit-learn, seaborn, matplotlib are installed on your python instance]
- Connect to the data source in PBI **phone_classification.xlsx**
- Select Transform Data to access Power Query Editor
- Access and view Column Quality and Distribution (features can be switched on in the View toolbar). Take note of the feature distribution, data gaps and any data processing requirements. Identify the label column (price range) which you will aim to predict using your classification model which will become your y.

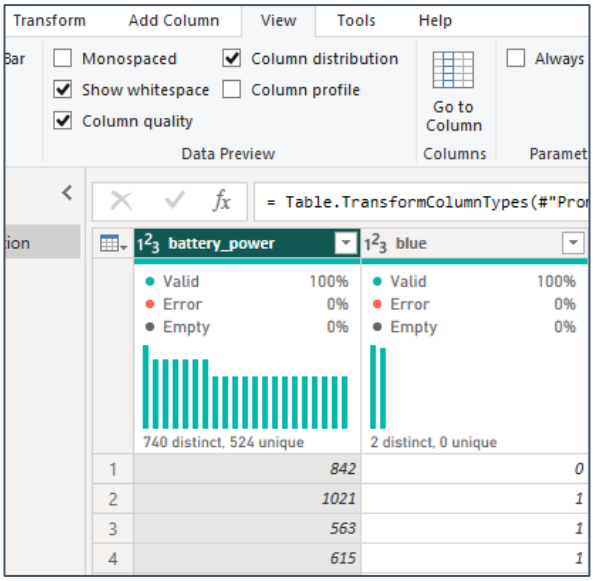

Fig 1. Using the View Tab in Power Query supports Data Evaluation

- Perform necessary data cleaning or data type transformations as appropriate to your data set using PowerBI native features.
- When satisfied with overall quality, duplicate the data and label the duplicate *DataFrames*. This is where you will primarily be processing data using Python scripts to create the data for your report.
- To begin using Python to process your data, select **Run Python script**

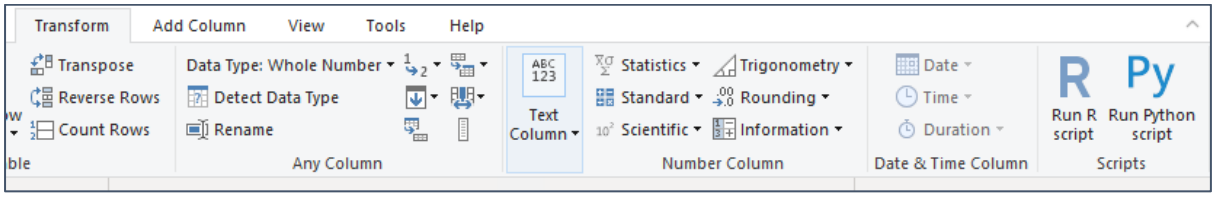

Fig 2. Python processing in Power Query

Data Processing with Python

- Assuming you already have a functioning model in python/ jupyter notebook, you can transfer much of the data processing model and evaluation script in your Jupyter notebook to this input box before selecting OK to run the script, noting the correct location on your computer where the python application is installed.

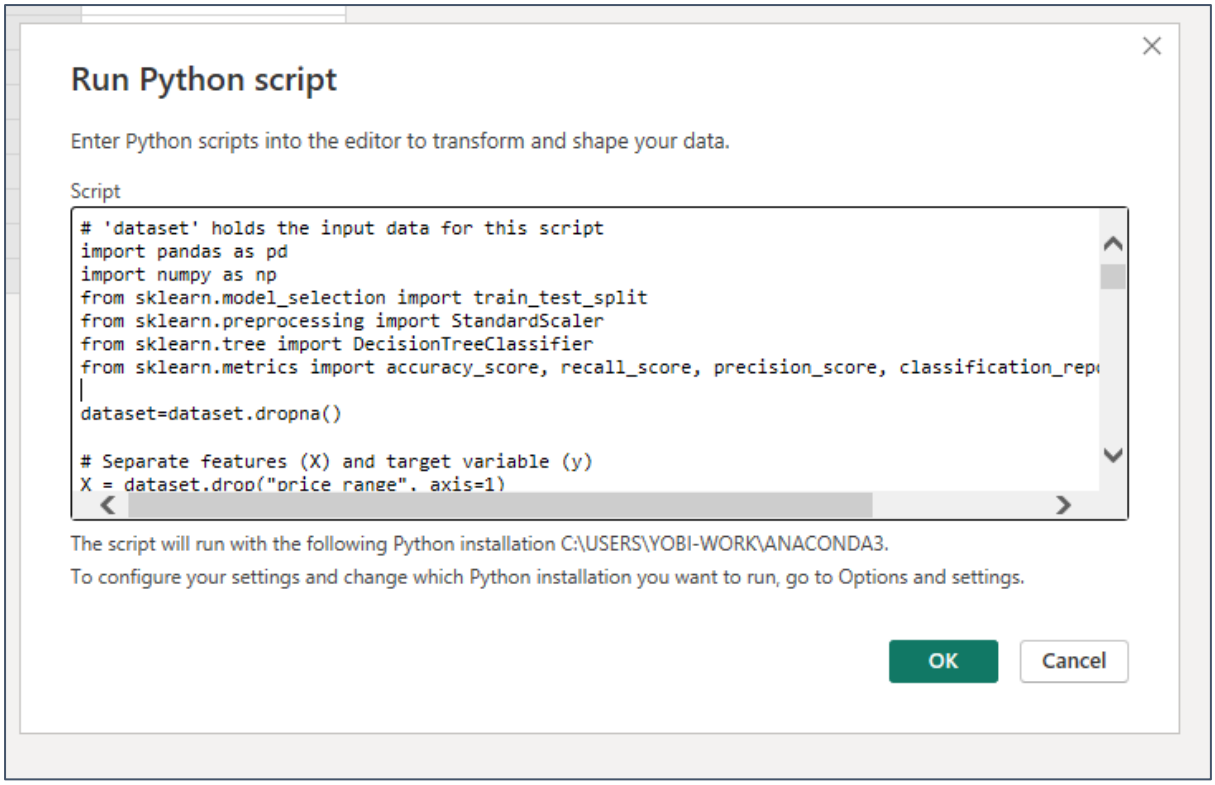

Fig 3. Python Script, run as a single block. Adapted code from a Python Notebook

- The script once run should result in a query in PBI which creates multiple data frames indicating the outputs of the python you have run.
- Following from the above, these are the data frames that you require in your workbook as PBI queries now to complete your report:
    - dataset (entire dataset is clean and validated)
    - train (scaled X_train and y_train_reset combined in a dataframe)
    - test (scaled X_test and y_test_reset combined in a dataframe)
    - most_informative_features (a simple attribute derived from the model)
    - conf_matrix (y_pred and y_test combined for evaluation)
    - Accuracy, prediction, recall (Model performance metrics combined in a df)

- Once you have completed the data processing, right click the target tables to save them as new queries in your workbook

DataFrames

1	conf_matrix	Table
2	dataset	Table
3	metric_df	Table
4	test_scaled	Table
5	train_scaled	Table
6	X	Table
7	X_test	Table
8	X_train	Table

Fig 4. Desired Output of **Dataframes**

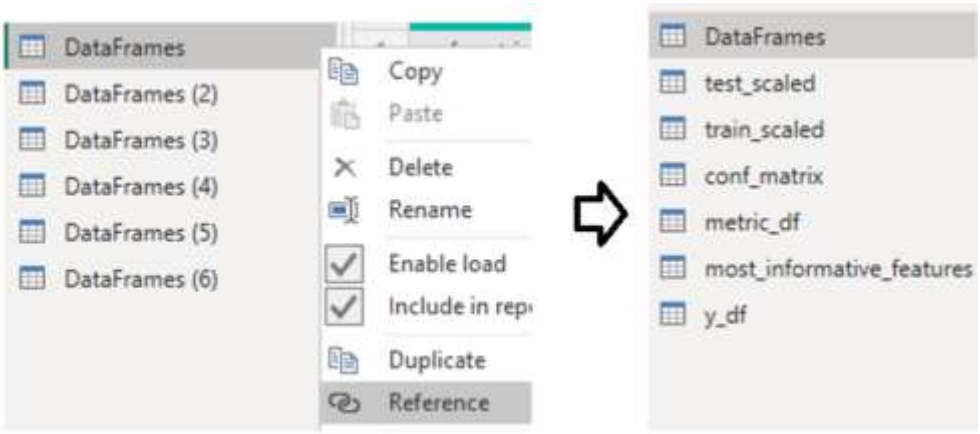

Fig 5. Reference the main Dataframe table, rename and select the corresponding table

Visualisations with Python

Once complete and you are satisfied with the dataframes you have processed. Go into the main PBI Desktop / report view and begin visualising the model and its performance metrics specifically using the Py tab under Visualisations

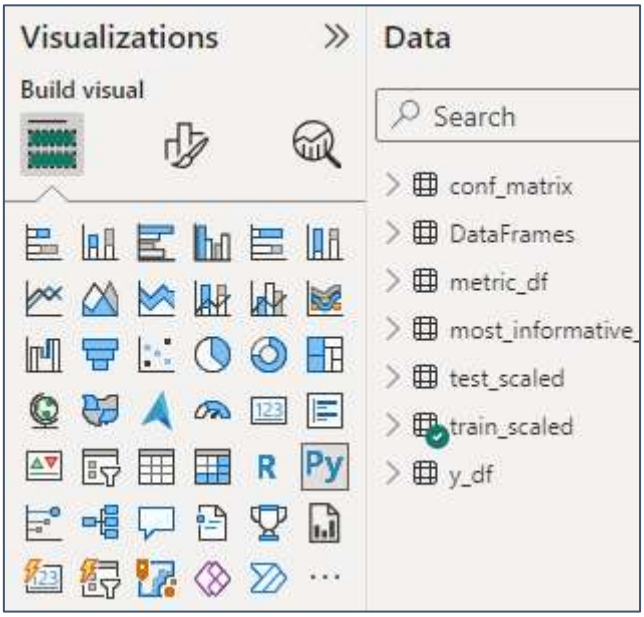

Fig 6. Select for Py visuals, there are some instances when visualising the model metrics, where you may opt for a simple PBI table.

- Adapt your python visualisation code from your notebook for the PBI Script editor. **Read the green Python script editor box** in PowerBI carefully to understand how to do this. Refer to the example workbook *PBI_DT_mobile_report.pbix* for guidance on this and

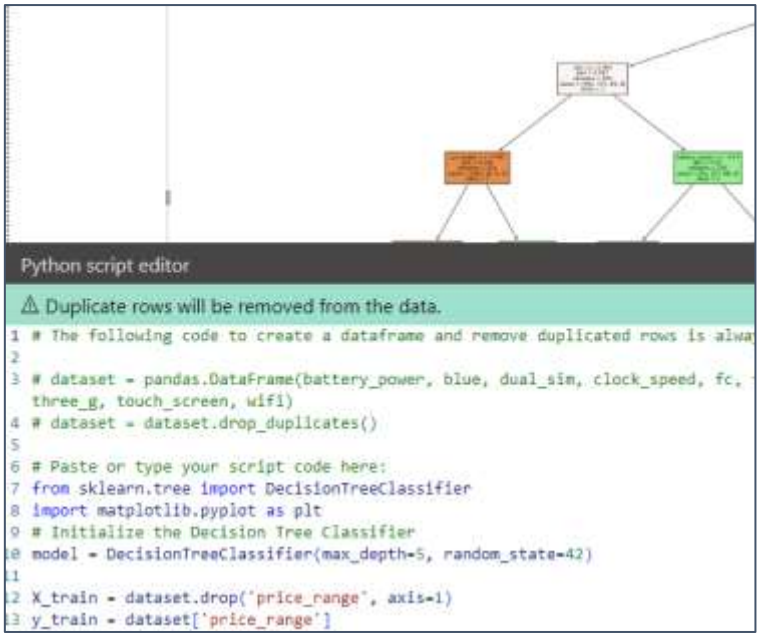

Fig 7. Python Script Editor will automatically manipulate your data.

Effective Communication

Experiment with *Figsize*, *page view* and *font size* to create a Decision Tree that is legible, even if you have zoom in to see what's written in the nodes. It is a dense visualisation and if we were to reduce the depth even further, we will have an overgeneralised model.

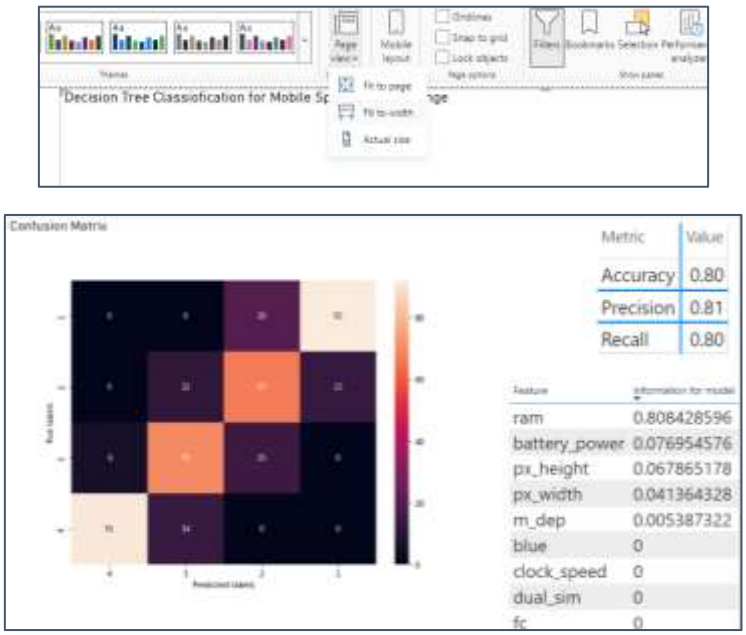

Fig 8. Report page view (top), example of Model metrics in both PBI Table and Python Script (Seaborn)