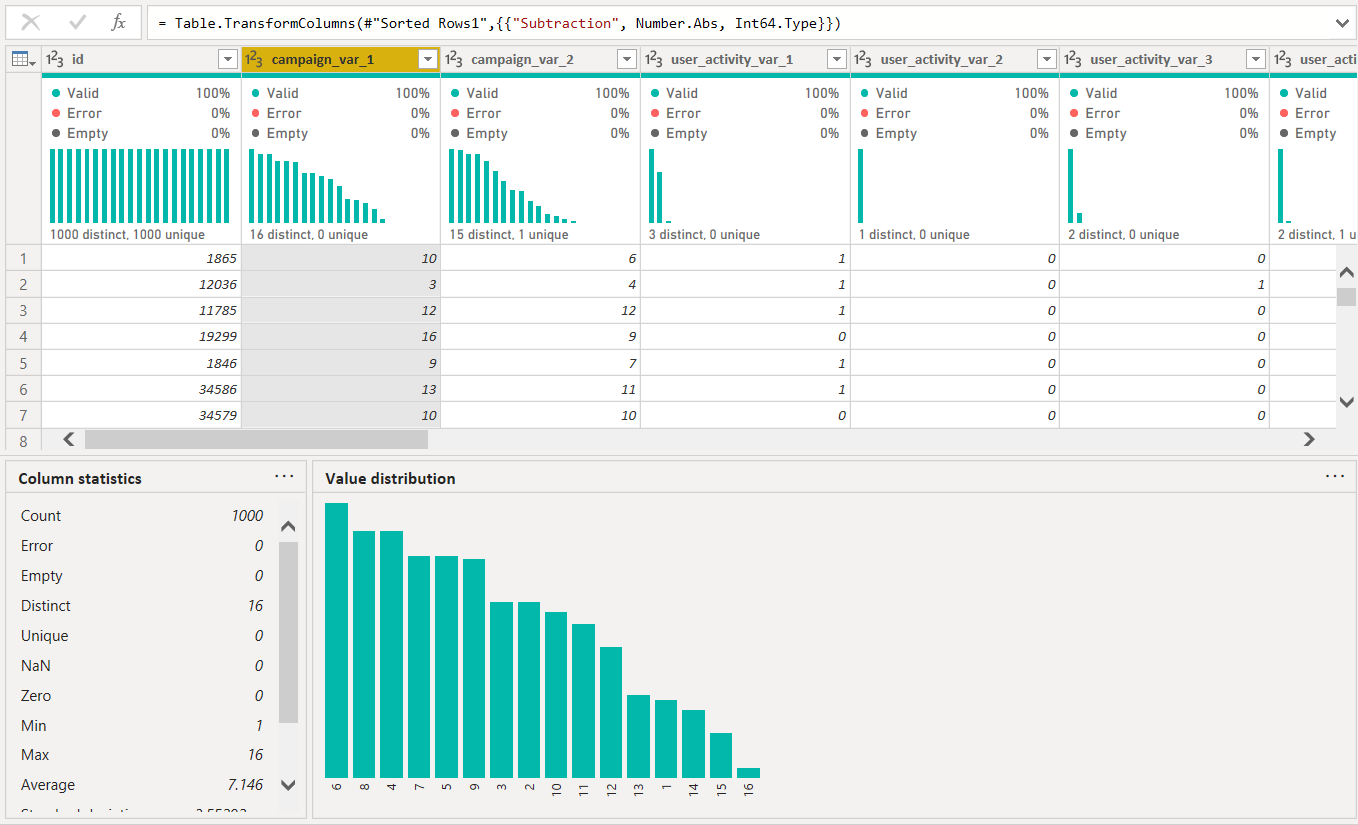
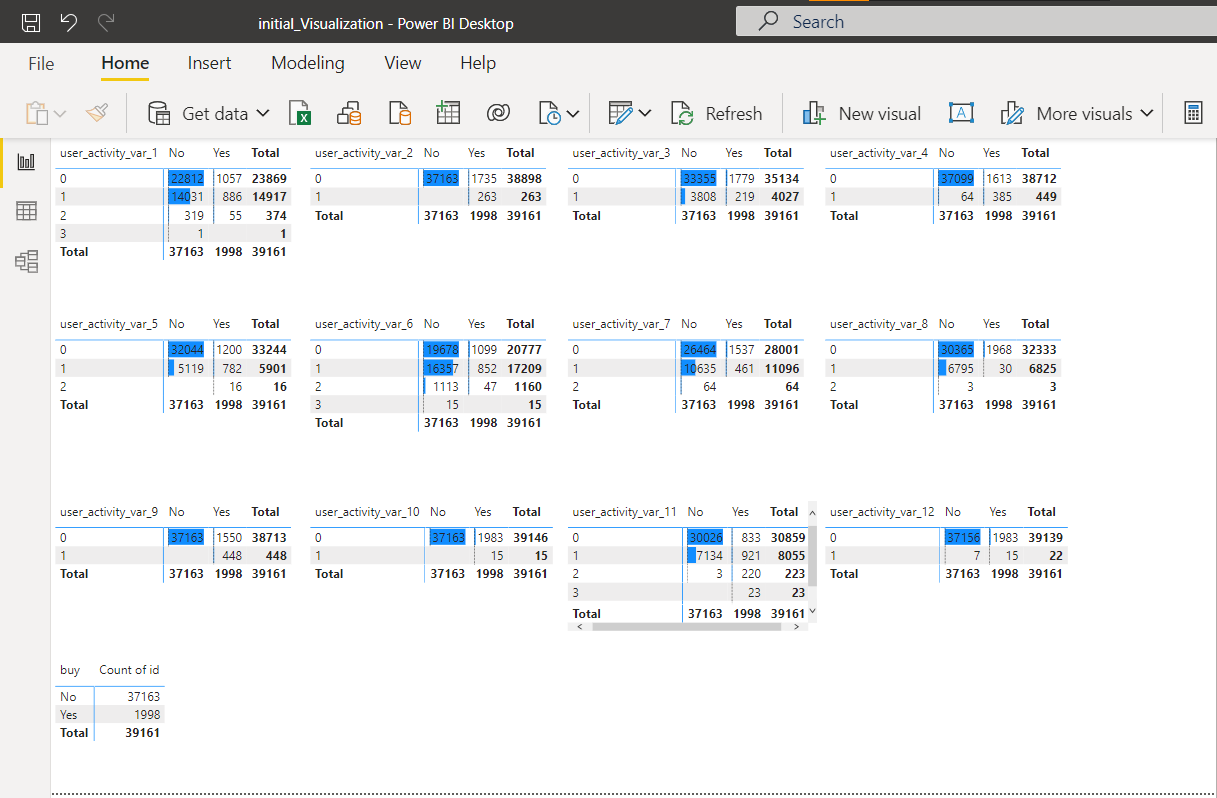
**1. APPROACH:**Used Power BI to analyse the data set with various tools and techniques. It helped me to get insights about the key influencing columns and the dataset. The approach involves these steps mentioned below.

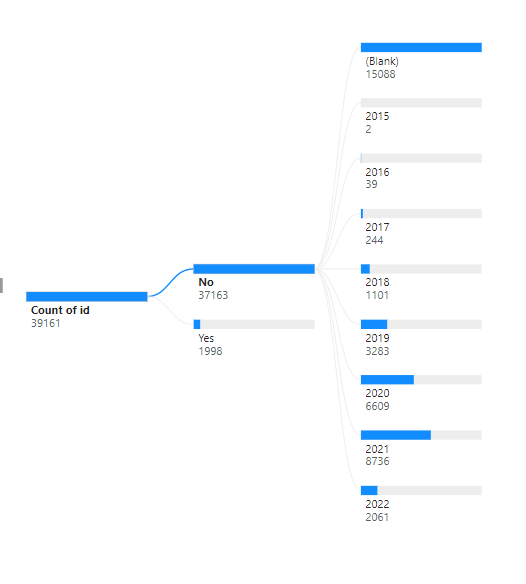
Checking the Distribution, Quality and Profile of each columns in the data set to get further insights



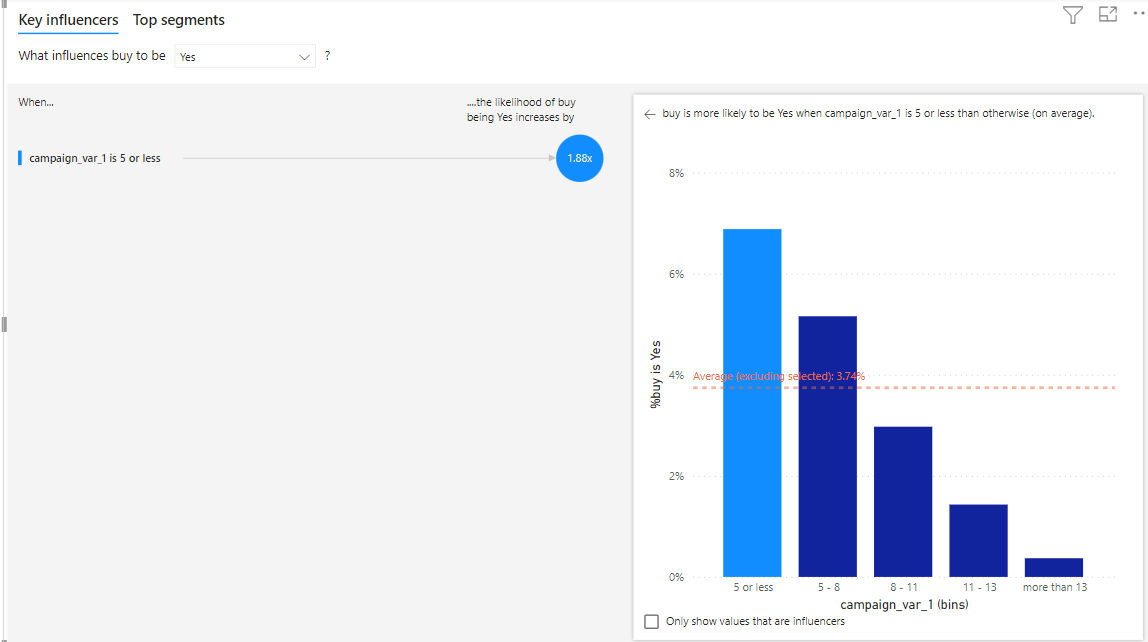
Checking the different values in each of the user\_activity\_variables to see if there is any key influence for the buy column from any one of these variables

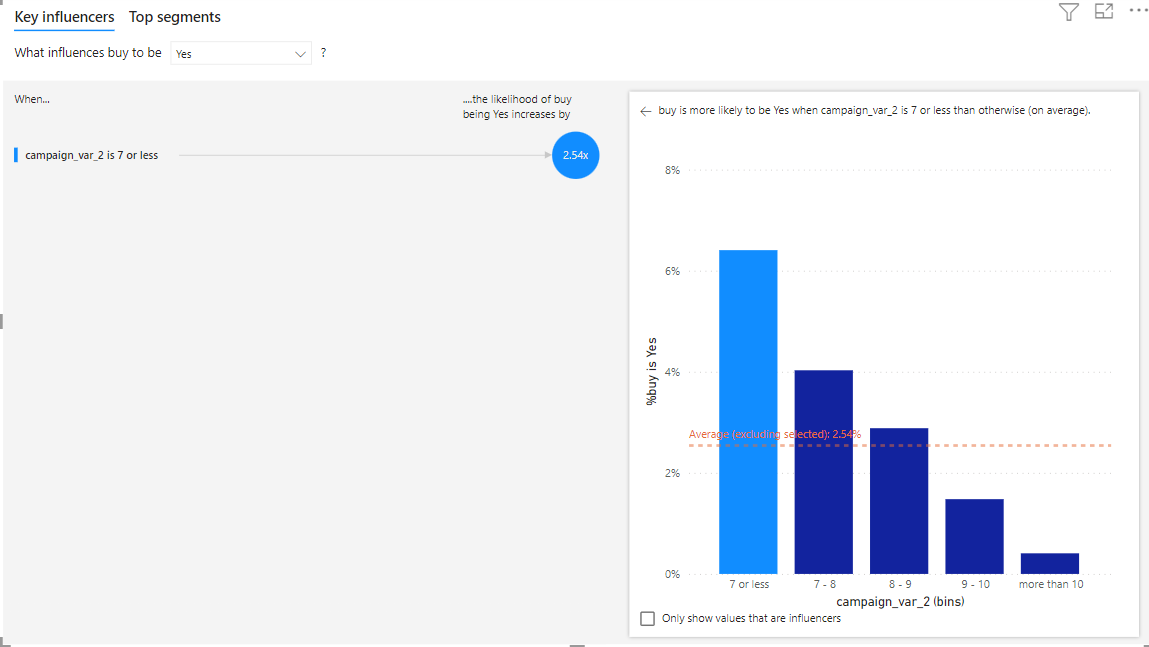


Created a Decomposition Tree to check verify the relation between the signed up year and number total number of buys in that year



Finding the Key Influencers for the dataset by using Power BI





**2. Data-Preprocessing / Feature Engineering:**

Increased the complexity of the model (increase variance) by creating duplicate features and added duplicate data points for the value buy = 1 since the data set had imbalanced data with buy being 0 in majority of the columns. Created multiple functions to process the data in the dataset

Signup function: Used to impute the null values as “No\_Value” and extract the year from the date in that data point

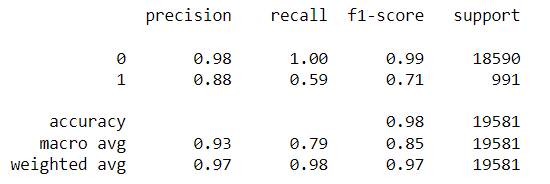
Camp\_var1: Used to group the campaign\_var\_1 into three different groups “less than 9”,”less than 13”,”1”

Camp\_var2: Used to group the campaign\_var\_2 into three different groups “less than 7”,”less than 10”,”1”

Buyer: Used to group the products\_purchased column into three types. 0 being null, anything less than 3 is 5 (Rare buyer) and anything more than or equal to 3 is 10 (Frequent buyer)

The Idea of increasing the complexity of the model by duplicating the columns/ features helped my model to perform well and I concluded that it could be helpful because of the biased dataset. Any simple model that throws a result as 0 for all data points will have at least 60% accuracy in this dataset as it is unbalanced with buy=0 overpowering buy=1. When we train a model with such data the model tends to have more bias than variance. Hence I’ve increased the complexity using duplicate features.

**3. Final Model:**

The classification report of the model looks like this

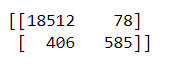
And the confusion matrix looks like this with

18512 – True Negatives

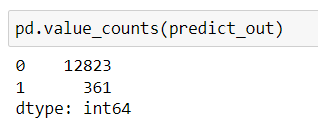
585 – True Positives

78 – False Positives

406 – False Negatives



And the total datapoints in the output has 12823 -1 and 361 -0



I’ve performed all the steps mentioned above and came to a conclusion with the score of 0.763231197771588 in the test data set.