# The create meaningful models for MediaMath project, the modeling work has been staged into Data Preparation, Candidate Model Selections. My project time is split into 60% and 40% between these two stages.

# Data Preparation:

1. Basic CSV file data cleaning. The CSV file contains a ‘Contextual\_Data’ data field, a dictionary describing the row structure. A data cleaning python procedure is created to remove the Contexual\_Data data field from the original CSV data file and also save the contextual\_data into a separate JSON file. After consulting with MediaMath contacts and manually checking myself, I decide not to include the contextual\_data data in the final study.
2. Feature selection.

First, we summarize the feature based on the type and the unique value counts in table below.

Table

Description automatically generated

From the data, one can quickly conclude that mm\_auction\_id is an id field and should not be used for the analysis. The imp\_timestamp should not be included in the data analysis because 1) it has a very big data size, 2) it corresponds to GMT time zone time and thus offers less usage to differentiate date part within the local time zone, and 3) some of its usage are already duplicated in other data fields such as day\_part. However, it is also worth pointing out that imp\_timestamp can be used in testing/training data selection, and this is indeed how the candidate models are trained and tested against.

Now we are ready to run feature selections on continuous (of data type float64) and discrete variables.

* 1. Continuous variable feature extraction. There is total of four columns that have float64 data type, and one of the data field, “exchange\_vcr” has a constant 0 value, and it is removed from the continuous variable input. In order to disallow redundant causal relationships among the input variables, for instance, x1=2\*x2, and thus knowing x1 is equivalent to knowing x2, the sample correlation is performed among these continuous random variables where no clear correlation has been identified. Therefore, all the rest columns are selected. The correlation chart can be found below. Table

     Description automatically generated with medium confidence
  2. Discrete categorical variable feature extraction. Since any categorical value can be potentially viewed as an extra parameter that one needs to calibrate, and each parameter would require a certain amount of data coverage for calibration, our goal is to reduce the number of columns that have lots of categorical values and extract their categorical values in a special way. Considering this idea, we can now separate the categorical variables extraction procedure into two steps.
     1. A non-greedy algorithm is used to pick categorical variables as it is without any modification. Under this algorithm, one would keep adding the categorical variables (starting from the categorical variable with the smallest number of categorical values; thus, the algorithm is called non greedy) until the total number of categorical values surpasses a threshold.
     2. The algorithm would not include columns 'overlapped\_brain\_pixel\_selections', 'site\_id', 'base\_domain', and they will be processed in the feature extraction enhancement section.

1. Feature extraction enhancement.
   1. We would add more continuous variables that are derived from 'overlapped\_brain\_pixel\_selections' field. This extraction is based on domain knowledge from MediaMath contact. According to them, if there is a valid 'overlapped\_brain\_pixel\_selections' field, it would incorporate browsing recency and frequency information in its content. A python extraction code is used to add recency and frequency fields into the column fields.
   2. As to the categorical variables, we would revisit the included columns and see whether certain reductions can be performed. The reduction is based on our target loss function has the form . Let’s assume we have only two categorical variable X\_1, X\_2, the loss function may indicate that we need to identify a function form f(X) with two variable X\_1 and X\_2 where each one can take any of its listed categorical values, however, if the P(X1=x\_1,X2=x\_2) =0, there is absolutely no need for us to include the specification of f(x\_1,x\_2) in our model selection. Therefore, we should only focus on the joined (x\_1,x\_2) set from the observation data set. This also means that we can combine both (X\_1,X\_2) into one categorical variable X1\_X2, and make this data set include all observed (x\_1,x\_2) pair. This approach would only makes sense if the categorical variable X\_1 and X\_2 are highly linked, i.e., knowing one variable would automatically allow one to know the other one. Thus we can combine two into 1. This pattern can be obviously seen between “device model” and “device manufacturer” columns. For instance, knowing device models such as ma\_AT&T:mo\_Calypso or ma\_AT&T:mo\_Maestro, we automatically know it is made by device manufacturer ma\_AT&T, and we should need to make the model to learn NOI from ma\_AT&T:mo\_Calypso or ma\_AT&T:mo\_Maestro. The combination saves lots of parameters and possible unnecessary cross training. Use the same example, there are total 984 device models, and 98 device manufacturers. If we need to predict f(device model, device manufacturers), we need to specify a NOI value for each combination which is a 96432 (98\*984) dimension problem. Now by observing, they are totally linked fields, we can now reduce the dimension of calibration to 984 which is a big saving. By applying the same methodology, we can also combine 6 more categorical variables in the same way.
   3. Reduce the size of the categorical variables on Domain/Site/Publish\_ID. Here I follow a similar idea as in 3b that is to identify the columns with the low dimension that are highly linked to Domain/Site/Publish\_ID. The idea here is that individual devices would usually have a default browser that may prefer certain websites over others. For instance, MSN can be the defaulted website for the windows browser. To prove this is the case, we plot the 2d cross-table heatmap between the base domain/device model. If the base domain is evenly distributed across device models, then we should see non-cold color (some yellow or green lines or dots in the heatmap, but instead, we have seen a solid blue chart. This means that Domain/device models are clustered into very few joined data points. We now successfully categorized all categorical variables using cross tab and meaningful total observation coverage. The number of parameters is about 10k, which is significantly smaller than the initial total size of 50k. This is about 80% reduction. Chart

      Description automatically generated
   4. Continuous log transformation. Some of the continuous variables have a very skewed distribution, and log transformation is applied to smooth it up.
2. NOI analysis. One key question one would like to ask is whether our loss function is well-designed on our data. A quick cluster-based study has shown three facts.
   1. The NOI is largely concentrated at 0 values. See violin chart 1
   2. For none 0 NOI, the NOI is not evenly distributed. See violin chart 2.
   3. For none 0 NOI, the cluster study shows the concentration in 3 clusters, and therefore we should see that our model has meaningful coverage in these clusters.

Chart

Description automatically generatedChart, rectangle, box and whisker chart

Description automatically generated

A picture containing chart

Description automatically generated

# Candidate Model Selections. (All calibrations are based on preprocessed output that is mentioned in previous section)

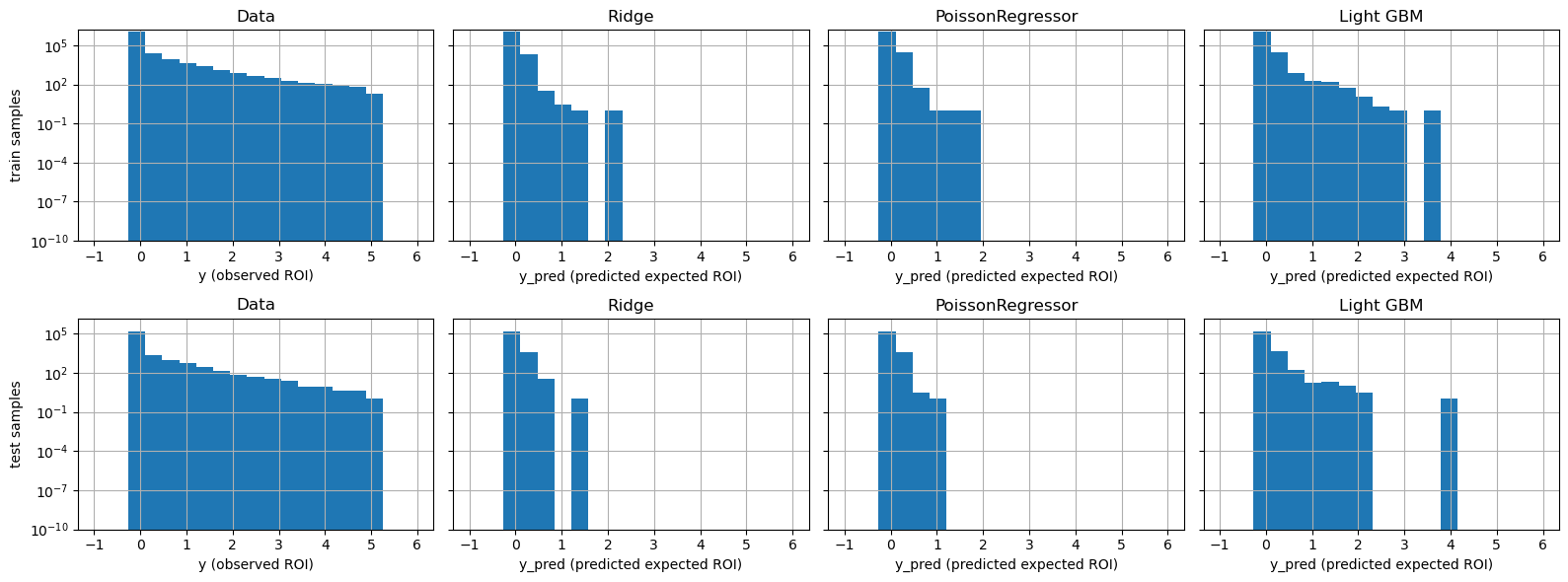
1. In candidate model selection, the training and testing data set are split between imp timestamp before 12/04 and after (include) 2022-12-04. The split is based on the fact that random split may accidentally include almost all 0 entries and thus make the test set representative. By using the time stamp to split, one can get better NOI presentation in the result.
2. Total 5 candidate models are used, and they are
   1. Zero mean model. The model forecasts everything to be 0. This is a benchmark model and was asked by the course adviser.
   2. Average bias (sample mean) model. This model would use the average NOI as the forecast. This is the true benchmark model we would like to compare.
   3. Simple Ridge Regression model. Given regression model would accidentally include a negative forecast, the deviance measure on both the testing and training data set would skip these points.
   4. GLM Poisson regression model.
   5. Ligh GBM Poisson Model.

The model performance summary is listed below. All RMSE is less than 0.18.

Table

Description automatically generated

1. It is clear that RMSE is not sufficient to differentiate the model. The 0 mean model is clearly not a good model, but it has a meaningful RMSE in the testing data set; however, its deviance is terribly bad. This means deviance is a good measure to help us differentiate the model. However, deviance also has its shortcoming since it can’t allow us to spot the tail deviance (when NOI is bigger than two and it is a meaningful value to catch according to cluster study). This result can be found in the chart below



1. Lastly, we would also like to run a quantile regression-type comparison to see whether prediction and observations agree in the quantile bins. The result shows LightGBM may overfit in the high NOI bin, though overall, it did the best job.

# Graphical user interface, application Description automatically generated

# Next Steps

1. Tune hyper-parameters for both Light GBM and Poisson GLM. The Poisson GLM is very sensitive to penalty term alpha specification. If it sets wrongly, one can see RMSE as high as 10!
2. IF hardware allows, I would like to test the xgBoost since it would allow more theoretical penalty term specification.
3. I want to use Up/down sampling to retrain the data so that low occurrence NOI has more observation in the calibration. This technique can be necessary for this study.
4. If time allows, I would like to search for better model selection ideas in both measures and charts besides what is already provided in this paper.