Sreya Dhar
Fraym.io Q&A

1. Why did you select to try the models you did? Did you consider adding any spatial models to your approach?

A: Multivariate Linear Regression, tree-based models, SVM, or KNN have been applied to the dataset. From the previous experience, I could gather that tree-based model should be performing well on the present dataset. The worst-case KNN models would be reliable in case CART models would not perform well in predictions. Thus, most of the general ML algorithms have been applied for the predictions.

Any primary spatial regression model like Spatially lagged exogenous/endogenous regressions have not been applied to the current dataset to discover the hidden pattern induced from spatial correlations on the response variable. It could be considered in the future scope of the study. Other than that, spatial visualization tools like *plotly* and *folium* have been applied to the dataset for exploratory data analysis.

2. Please contextualize your accuracy score. What does it mean to be 57% accurate when it comes to median spend? Are the differences in accuracy significant?

A: The accuracy (%) has been derived by subtracting percentage value of mean normalized absolute error from 100. Eq.1 has been adopted to calculate the accuracies on the train and test sets for various models.

$$\text{Accuracy (\%)} = 100 - \left[\frac{1}{n}\sum_n\left\{\frac{(y-\hat{y})}{y}\right\} \times 100\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{ Eq. 1}$$

Where, n = no. of observations/rows, $y = target\ value, \hat{y} = predicted\ value.$

The differences in accuracies are significant for train and test sets and different ML algorithms that have been applied for predicting the median expenditures. For example, the accuracies in the test set are 57% and 35% for XGBoost and SVM (kernel=*radian bias*) analysis, respectively.

3. Several of the accuracy scores differ between the train and test sets. At what point do you worry about overfitting?

A: In most of the ML algorithms, the model is hyper-parameter tuned to get the best optimal results on the test set. When the same model has been validated on the test set, the accuracy percentage dropped from the training accuracy. Two types of scenarios have been observed in these cases:

1. The hyper-tuned model is performing best on train and test sets simultaneously (maintaining a similar loss/accuracy pattern). Thus, yielding the optimal error on train and test sets irrespective of the differences in accuracy scores (most of the tree-based models).
2. The best of hyperparameters is not performing equally on train and test sets (KNN model). That means the optimal error of the train and test set are not co-occurring, and the model is performing best with different groups of hyper-parameters on train and test sets. In that case, overfitting would be a sensitive issue and needs to be handled carefully maintaining a balance between the predictions from both sets. For example, in the KNN model for k=3, the prediction accuracy for the train and test sets are comparatively similar; thus, k=3 has been chosen for optimal k for prediction. Though, the optimal training error is not occurring for k=3 in the KNN model.

When the pattern of training and test error curves start to diverge from each other (like, training accuracy starts to increase and test accuracy drops), that would be the culminating point to stop the iterations from further overfitting.