Feature Engineering for Sparse Demand Prediction

Masters Student Paper Competition Submission

Hsu Hsiao Yu, Robyn Campbell, Stefanie Walsh, Zinnia Arshad, Matthew A. Lanham
Purdue University, Department of Management, 403 W. State Street, West Lafayette, IN 47907
hsu230@purdue.edu; campb170@purdue.edu; walsh88@purdue.edu; zarshad@purdue.edu; lanhamm@purdue.edu

ABSTRACT

This study provides feature engineering recommendations for predictive modelers, data scientists, and analytics practitioners on how to improve demand forecasts for sparsely demanded specialized products. The motivation for this study is that there are many suggestions on methodologies for problem types, and general feature engineering ideas, but there is no large-scale study to date that provides an in-depth empirical investigation of feature engineering approaches and their associated predictive gains when trying to predict sparse demand – which is one of the most challenging prediction problem classes one can encounter in practice. In collaboration with a national retailer, we develop predictive models to predict demand for 47k+ products where 26k of them have less than five units sold in a year. Problems such as these are common in medicines, specialty products, and auto and military spares. What is novel about our study is that we run thousands of feature engineering experiments to identify where we see cross-validated predictive gains for a set of common predictive modeling algorithms. For example, various categorical encoding schemes (one-hot, frequency, label, hash, and target encodings), various scaling/transformation techniques, outlier handling for numeric data types, as well as variable fusion strategies such as interactions, powers, and ratios. This work is unique as much of the literature focuses on predicting product demand with larger quantities (non-sparse demand), supervised learning methods, or general feature engineering ideas. We show how to implement a similar large-scale feature engineering study, provide empirical insights of where we achieved noticeable gains, and why what we realized with our data could likely work for practitioners faced with a sparse demand prediction problem.

Keywords: Feature Engineering, Sparse Demand, Encodings, Transformations, Supervised Learning

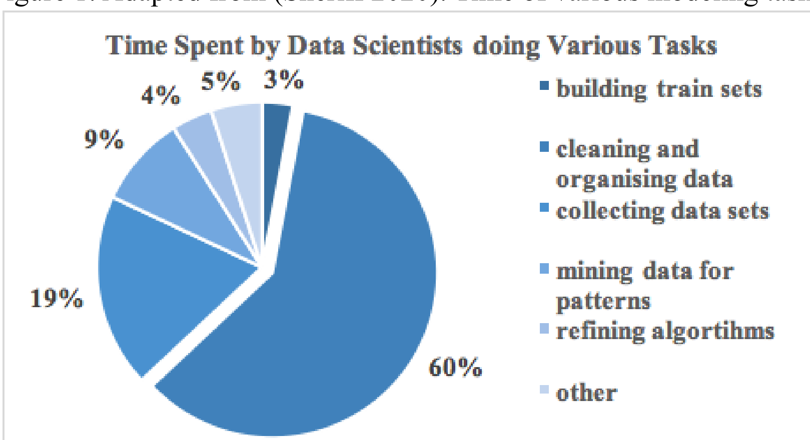
INTRODUCTION

Any seasoned modeler knows that predictive modeling is a process and there are many possibilities on how one might clean, pre-process, and format their data prior to training a model. Additional complexity often arises based on the problem characteristics (e.g., temporal response, intermittent demand, sparse demand, etc.) that can make identifying the signal from the noise even more challenging. In the field there is often some discussion of "the art with the science" on best practices to perform to achieve model accuracy good enough to support major business decisions. While there are many suggestions on methodologies for problem types, and general feature engineering ideas, there is no large-scale study to date that provides an in-depth empirical investigation of feature engineering approaches and their associated predictive gains when trying to predict sparse demand – which is one of the most challenging prediction problem classes one can encounter in practice.

Feature Engineering (FE) uses various data mining techniques to derive important features from raw data using domain knowledge. These features will in turn either allow the algorithm used to capture the signal or not, which will impact the predictive performance of the model.

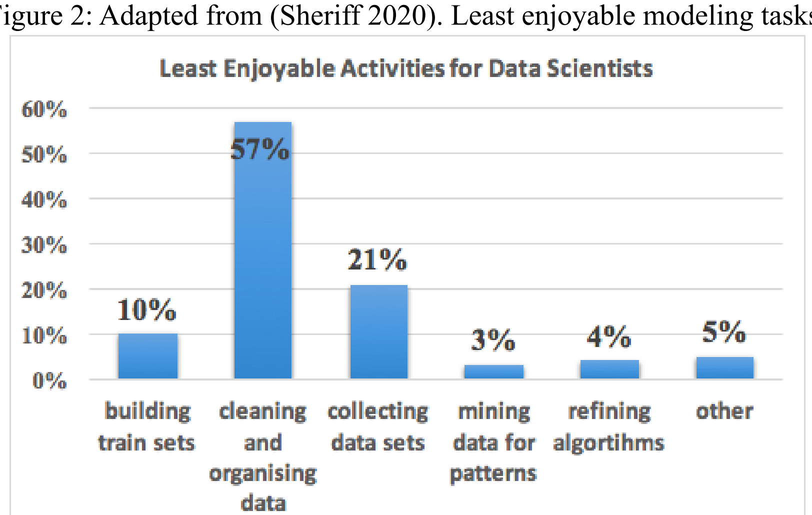
Multiple reasons make FE an important and growing field in the analytics sphere. The choice of features can impact how complex your models will be and FE leads to simpler models which will consequently reduce the costs and time taken by each model to run yet increasing accuracy (Xie 2020). Feature engineering not only helps to significantly reduce the feature space dimension but also contributes to construct a new feature space where data are more sparsely distributed, therefore ML models are easier to develop (Zhang, Cao et al. 2018). Usage of FE can significantly reduce the time taken for data preparation as the sets organized post FE will optimal and will contain all the essential features which are all essential to solve the business problem at hand (Press 2016). Data scientists spend 80% of their time preparing and collecting data out of which 76% find this to be the least enjoyable part of their job. Figure 1 shows the estimated time data scientists typically spend on various modeling tasks.

Figure 1: Adapted from (Sheriff 2020). Time of various modeling tasks.



According to Sheriff (2020), nearly three fifths of the least enjoyable time spent in modeling process among practitioners is cleaning and organizing their data as depicted in Figure 2.

Figure 2: Adapted from (Sheriff 2020). Least enjoyable modeling tasks.



Accuracy of models can be improved if FE is employed. According to a study done by Gartner on a cancer prediction dataset, various models (e.g., logistic regression, Decision tree, SVM, MLP) saw an improvement in the overall algorithm's accuracy once FE was applied on the raw data (Gartner 2019). This area of study needs more research in terms of automation of FE to keep with the pace of ever-changing business scenarios. Due to operational gap and impaired communication between different teams creating features to use in production environments it can be a prolonged process, hence more training and research needs to be invested in this area to make it more adaptable and agile.

In our study, the motivating business problem this paper focuses on is to improve demand forecast for products that are specialized with sparse demand, by delineating the most important features used by the company and predicting sensitivity of these features to the forecast prediction. The research questions that this paper will address are the following: How accurately can the firm predict demand of specialized products? Which variables are important in improving demand forecasts? How sensitive are these variables to various changes to get a more robust forecast?

Our research focuses on these questions as the literature is scant on demand forecasts for specialized products which are sold only a few times a year. Companies which have very diverse demand for various products will find value in this work since some of their products are sold in thousands and some are sold irregularly. For example, pharmaceutical/medicines, specialty products, and auto and military spares. This paper experiments with various FE techniques for a range of model flexibility (e.g., logistic regression, gradient boosting, and neural networks) to predict demand of the sparse products.

The rest of the paper is organized as follows. We first provide a review of the literature on various criteria and methods used for feature selection. Next, we discuss the data used for our experiments. The next section provides our methodological design and experiments performed. In the Results section we discuss what we observed from thousands of experiments and the conclusions we provided our industry collaborator.

LITERATURE REVIEW

Feature engineering has been an area of research widely for predictive modeling with various types of machine learning models being tested to evaluate their performance against multiple FE techniques. This has been established by (Heaton 2016) as it was concluded that neural networks and support vector machines generally benefit from the same types of engineered features; similarly, random forests and gradient boosting machines also generally benefit from the same set of engineered features. The former made the counts, differences, powers, and rational polynomial engineered features all relatively easy to synthesize and the latter failed to synthesize the counts and ratio (difference features). They conclude that ensemble methods generally perform better than individual models. They suggest that future research is needed to explore multiple other FE techniques with a plethora of machine learning models. The focus should be shifted to FE methods that are made up of multiple input features.

FE can be time-consuming and less likely to led to outcomes a modeler might be targeting if based solely on a data scientist, hence, it is often suggested to have domain expertise available whenever possible. Khurana, Samulowitz et al. (2018) propose a novel approach to automate FE using reinforced learning. They conclude that their proposed model can reduce the error rate (25% by median) and save hours of team member workload. However, they suggest further research is required to improve the efficiency of the system and extending this framework to various other FE techniques.

Machine learning models have been used expansively to forecast demand. However, not much research is have be invested in sparse demand. Khan and Ahmad (2004) propose the preprocessing of data using FE to improve forecasting for an energy dataset with a primary focus on indicating several ways to missing values imputation and model pipelines. They conclude that the ensemble methods XGBoost, CatBoost, and random forest models performs better than ARIMA and neural network-based (e.g., LSTM) models. They did this study on a dataset with large gaps and found that in comparison to the model with missing data, the proposed hybrid model with imputed data performs much better.

Rawat and Khemchandani (2017) propose a set of methodologies, techniques, and tools for FE to improve accuracy of classifiers on unseen data. They conclude that FE simplified the feature selection process and led to a higher classification accuracy compared to non-engineered features. However, further research from both these papers state that more work is needed in this field in terms of using different algorithms like recurrent neural network and using multiple other FE techniques. This provides the groundwork for our paper as we will be focusing on multiple other FE techniques and models for sparse demand prediction.

We summarize our findings in Tables 1 and 2.

Table 1: Key papers and identified research gaps

Study	**Insights**	**Research Gap**
(Heaton 2016)	Conclude that ensemble methods generally perform better than individual models.	Future research is needed to explore multiple other FE techniques with a plethora of machine learning models. The focus should be shifted to FE methods that are made up of multiple input features.
(Khurana, Samulowitz et al. 2018)	Their proposed model can reduce the error rate (25% by median) and save hours of human workload.	Further research is needed to improve the efficiency of the system and extending this framework to various other FE techniques.
(Waqas Khan, Byun et al. 2020)	Ensemble method using XGBoost, CatBoost, and random forest model performs better than ARIMA and neural network-based models.	They did this study on a dataset with large gaps and found that in comparison to the model with missing data, the proposed hybrid model with imputed data performs much better. Our study is a different case study of this.
(Rawat and Khemchandani 2017)	FE simplified feature selection process to obtain a higher classification accuracy compared to non-engineered features.	More work is needed in terms of using different algorithms with FE techniques. This provides the groundwork for our paper as we will be focusing on multiple other FE techniques and models for sparse demand prediction.

Table 2: Relation of our study to other academic papers

Study	***Paper Aspects***				
	Data Imputation	Feature Engineering	Machine Learning	Reinforcement Learning	Demand Forecast
(Heaton 2016)		V	V		
(Khurana, Samulowitz et al. 2018)		V		V	
(Waqas Khan, Byun et al. 2020)	V		V		V
(Rawat and Khemchandani 2017)		V	V		
Our study		V	V		V

DATA

In collaboration with a national retailer, we were provided data related to their demand prediction process, features used in modeling, and their observed demand response. The novelty of this data was inscribed in its sporadic nature. The dataset was on a SKU-level and another one was related to SKU demand per store location. The former had relatively consistent data points without many null values because of its higher level of aggregation. However, at the SKU level we observed many zero purchases over their operational window. This indicated that multiple products were only selling a few units per planning cycle on an aggregated national level. Thus, the dataset was split into SKU demand at a store level for products that sold less than or equal to five times in the previous planning cycle and products that were sold greater than five. Table 3 provides a general description of the data we used in this study.

Table 3: Data used in study

Variable	Type	Description
Sales	Numeric	Product sales ($) for multiple planning cycles
Product description	categorical	Overview of the product ID, category, base product group and lifecycle
Demographics	Numeric	Population estimate, household income, percentage white collar/blue collar workers, age of their consumers, road quality index and number of registered businesses divided by region
Product loss	Numeric	The number of failure sales, wait time and products lost in the process of selling

METHODOLOGY

Our predictive modeling approach is divided by three parameters - national level data vs store level data with sparse demand and with continuous demand. We ran several different types of models on each of these data sets. To account for memory issues due to the size of the data, sampling was done at the store-level data. The cross-validation design we employed a validation set approach using a 70/30 train and test set partition to ensure that the training dataset included all the possible patterns used for the problem. Moreover, a good ratio of test data provides for a more robust estimate of the error rate. Although each data set has its own models, we used various statistical evaluation metrics (e.g., R-squared, RMSE, MAPE and MAE) to evaluate our models. We did this to benchmark the strengths and weaknesses of the experiments as these measures each have their own pros and cons. To prepare the data, rare label encoding was employed to ensure that both the training and test datasets had the same categories as there was a significant number of unique categories or factor levels in multiple variables. Experiments were run under several conditions, including categorical encoding, numeric encoding, transformations, outlier removal procedures, and scaling techniques. Each experiment was a different combination of those four options.

Figure 1: Study methodology

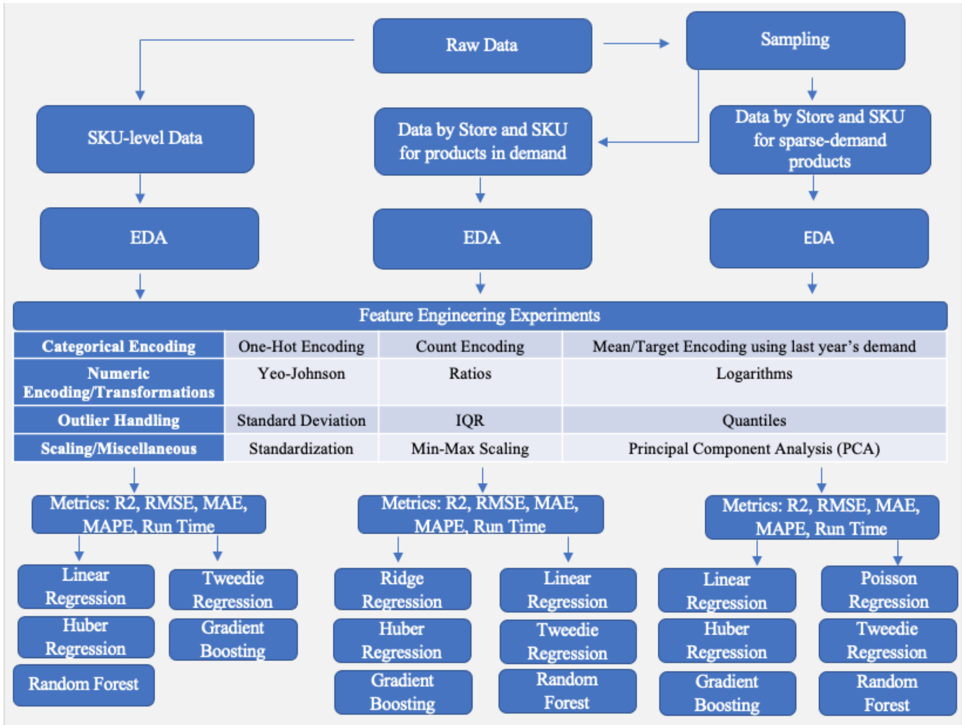


Figure 2 provides a pseudocode outline of our experimental design used in this study. Practically, this design is useful as an experimenter could add to their list of machine learning algorithms and FE techniques, they would want to try without having to do ad hoc analyses.

Figure 2: Pseudocode

```
df = Data
df fill missing with "NA"
df drop variable with 1 value
df drop current year variables
X_train, X_test, y_train, y_test = split df into train and test data sets 30% test
pipe = Pipeline([Make negative sales equal to zero, Make rare categories one category called "Rare", Drop nearly
constant variables, Drop highly correlated variables])
fit pipe to X_train, y_train
transform X_train, X_test
mod = list of machine learning models
num = list of numeric transformations
cat = list of categorical encoders
outlier = list of outlier detection and capping
scale = list of variable scaling
d = empty data frame for feature importance scores
CHARACTERISTICs: empty lists of model elements used and model scores
# For loop
for m in mod:
    for n in num:
```

```
    for c in cat:
        for o in outlier:
            for s in scale:
                X_tr = copy of X_train
                X_te = copy of X_test
                pipe = Pipeline([c,n,o, drop highly correlated features])
                fit pipe to X_tr, y_train
                transform X_tr, X_te
                names = list of transformed X_tr variables
                if use scaler:
                    Fit scaler to X_tr, y_train
                    transform X_tr, X_te
            model = m
            fit model to X_tr, y_train
            y_tr_pred = model predictions for X_tr
            y_te_pred = model predictions for X_te
            if model is tree based:
                importance = model feature importance
                d2 = data frame of importance with names as columns
                d = concatenate d and d2
            else model is linear:
                importance = model feature p-values
                d2 = data frame of importance with names as columns
                d = concatenate d and d2
          Append CHARACTERISTICS respectively
results = data frame of saved CHARACTERISTICS
final = join d to results
Save final
```

We have used six different types of models. Tweedie Regression was used on the SKU level dataset. It allowed us to deal with non-negative highly right-skewed data as well as symmetric and heavy tailed data. It can also handle continuous data with probability mass at zero. Huber regression was used for the national data by SKU as well as the sparse demand store-level data. This regression technique was considered because of its robustness to outliers by providing an alternative loss function to the typical least-squares method. Its loss function provides similar penalties for data points with small residuals, but on larger residuals, it gives a lower penalty by increases linearly rather than quadratically. Ridge regression is used by store level data with continuous demand. It helps avoid overfitting, performs quite well with large multivariate datasets with large numbers of predictors because it does not require unbiased estimators. In more detail, ridge regression regularizes or shrinks coefficients of variables that are the least influential to the model, allowing it to build complex models and avoid over-fitting at the same time. In addition, because of this shrinking effect on the coefficients, ridge becomes useful when there is multicollinearity in the features. Linear Regression (OLS) is used in two data sets (of store level data) as a baseline to check the accuracy of the prediction. This was used due to its simple and easy of implementation. However, we used other models alongside it since its prone to underfitting and sensitive to outliers. Poisson Regression was initially considered a potential model to use for our sparse demand data set of store level data to ensure that the 0's in the data can be handled by the model. However, we observed early on that it did not perform well and thus removed it from most of our experimental runs. Gradient Boosting was used in three data sets (of store level data) as it was a more accurate model that usually has high prediction accuracy. Moreover, it handles missing data and reduces the need for data pre-processing, offering much higher flexibility. However, it may cause overfitting and is computationally expensive. Random Forest was utilized for the national level data. The benefits of Random Forest are apparent in its ability to handle large data sets as well as the speed of the algorithm which was necessary when running thousands of experiments (Kurama 2020).

RESULTS

We ran over 2000 experimental feature engineering combinations with the goal of trying to generalize what really works best for this clients data set and modeling goals. Doing all these combinations allowed us to empirically estimate which techniques really worked for them. First, we found minimal evidence that count or mean encoding improved model performance, but both ran much faster than one-hot encoding for an equal result as seen in Figure 1. For the SKU-level dataset specifically, one-hot encoding was slightly slower in most cases, until we ran Random Forest where we saw one-hot encoding significantly slow down run time, but it yielded the highest median R^2 result. As for the numeric transformations seen in Figure 2, the experiments showed that for more complex machine learning models, Yeo Johnson, a transformation used to make numeric variables more Gaussian, improved performance while for linear regression, ratio generation held the highest median result. We found support for this finding as around 50% of the ratio combinations generated were significant, compared to 16% for tweedie and 5% for the other models. This finding held across all datasets.

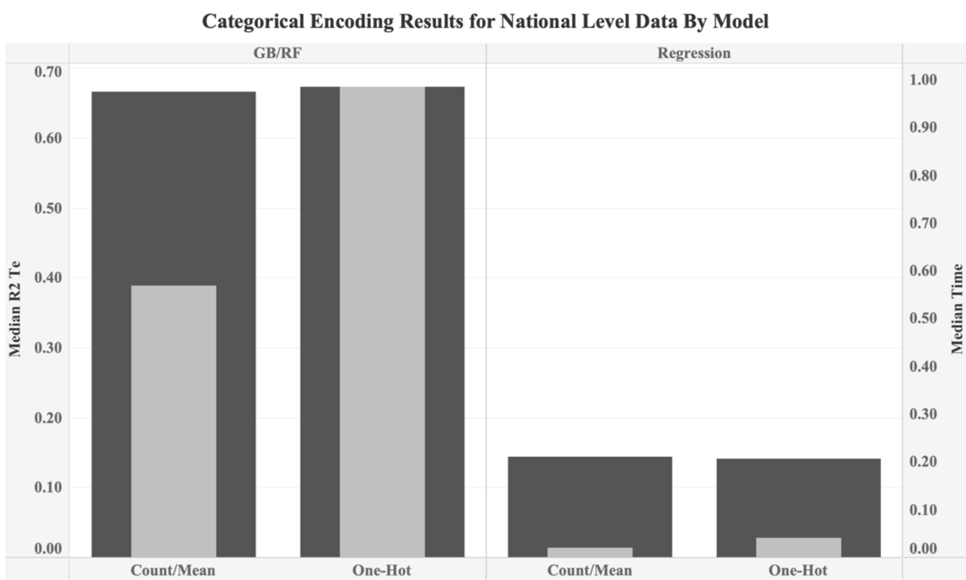


Figure 1.Categorical Encoding Results for SKU-Level Data By Model

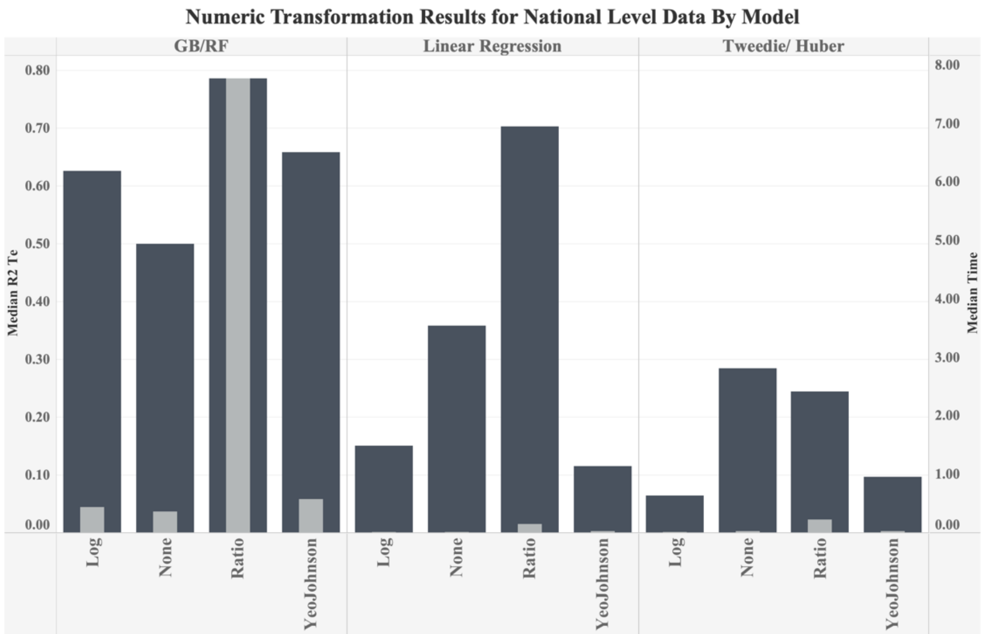


Figure 2. Numeric Transformation Results for SKU-Level Data By Model

For the SKU by Store Data for Continuous Demand products, we found that for Gradient Boosting, Ratio Generation actually created the best outcome, but it took much longer to run. For regression models, no transformation yielded the highest R2 values. In this second dataset, we do see a pattern for outlier removal in which gradient boosting models benefited from Outlier Removal using a Gaussian approach where regression models benefited from no outlier removal.

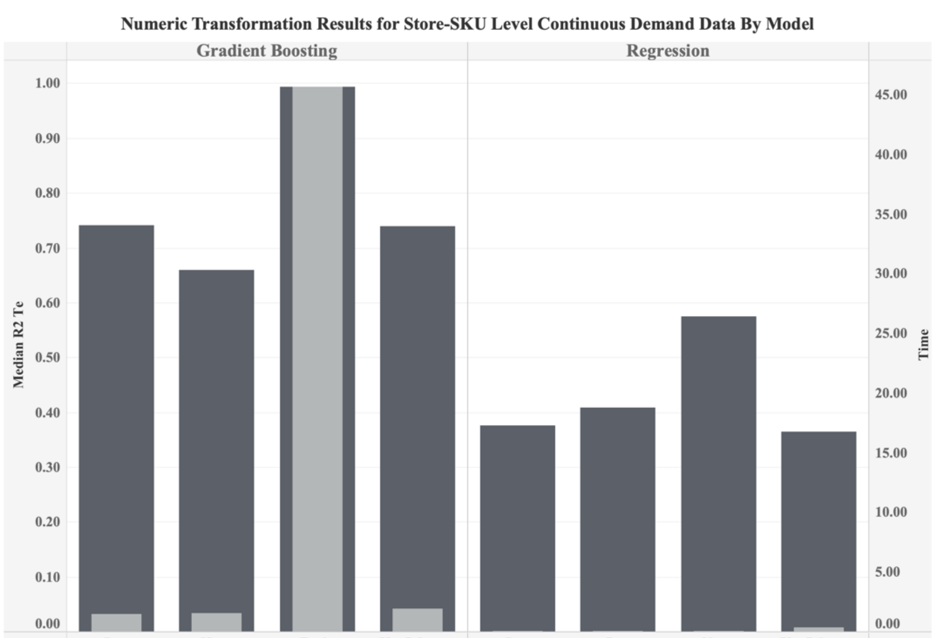


Figure 3. Numeric Transformations for Store-SKU Level Continuous Demand Data By Model

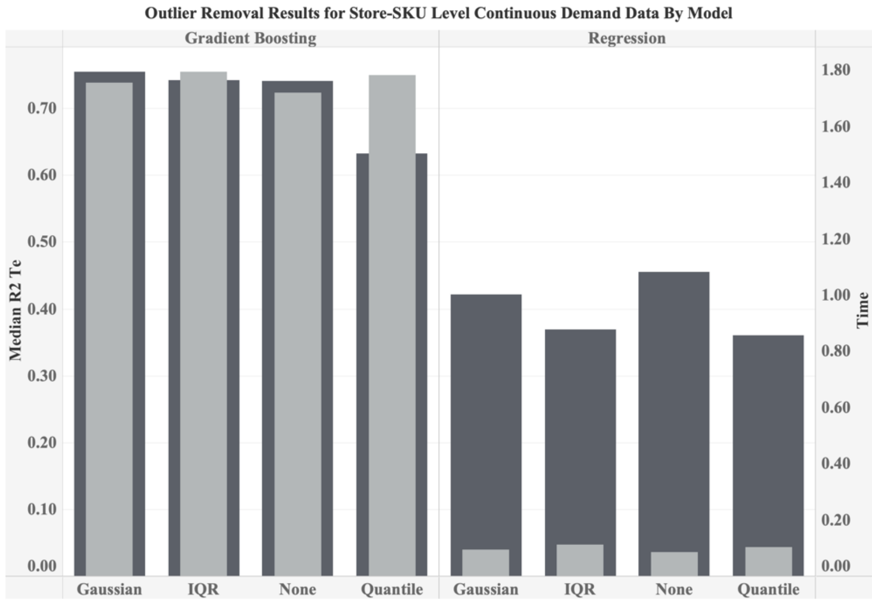


Figure 4. Outlier Removal Results for Store-SKU Level Continuous Demand Data By Model

Finally, for the SKU by Store level data for Sparse Demand products, we saw that regression models like Tweedie and Huber improved with log transformations while in linear regression, no variable transformation performed the best. Finally, we see a unique pattern in outlier removal and scaling results for spare demand data. Specifically, for gradient boosting, outlier removal and scaling had no effect. For linear regression, not removing outliers led to better performance while scaling had no effect. For Huber and Tweedie regression, outlier removal had no effect, but z-score standardization did improve performance.

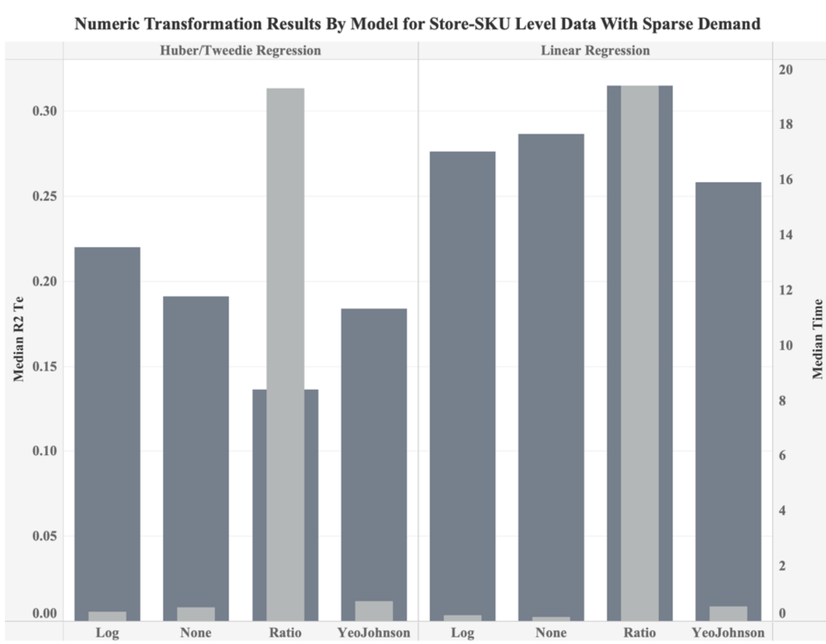


Figure 5. Numeric Transformation Results By Model for Store-SKU Level Data With Sparse Demand

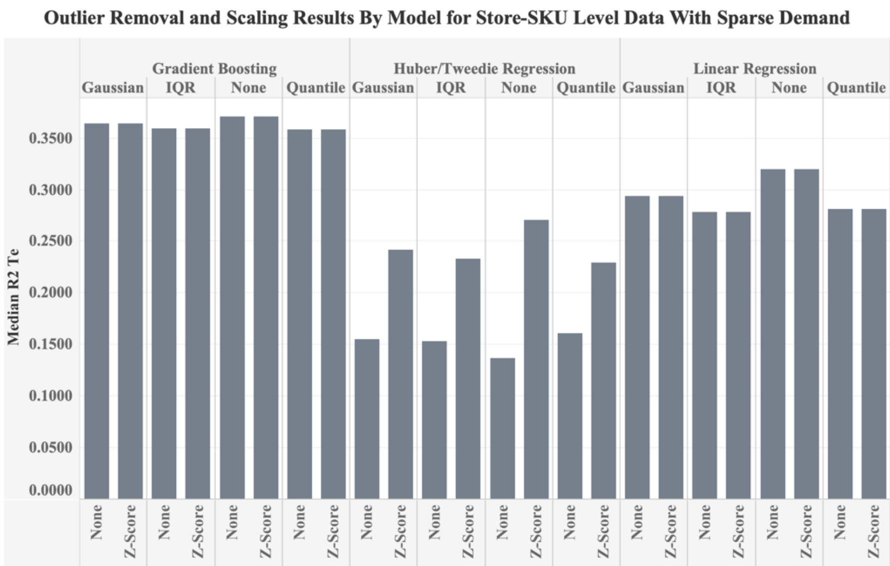


Figure 6. Outlier Removal and Scaling Results for Store-SKU Level Data With Sparse Demand

CONCLUSIONS

In this paper we have tried to resolve the problem faced by businesses when accurately predicting demand of products with sparse sales in different areas nationwide. The importance of this problem can be found in the commonality of this issue faced by multiple car part businesses and hence our solution can help them save costs by decreased runtime, more efficient models and more free time at the hand of their data scientists to fine-tune their models.

The conclusions that we drew from our analysis are that one-hot encoding consistently took longer to run and process than count encoding or mean encoding for equal results. Secondly, running more complex models like gradient boosting or random forest significantly increased run time. Thirdly, regression models are significantly more likely to have significant ratios than gradient boosting or random forest models. Thus, making it best numeric method to increase Linear Regression accuracy. Fourthly, scaling and outlier removal did not impact SKU-Level data. Lastly, when using alternative regression models utilize Z-score scaling to boost R2.

Our analysis can help the client decrease the running time for demand forecast using feature engineering algorithm by 20% which leads to cost saving on their computational as well as HR front. Secondly, they can develop dynamic experiments to make modeling more flexible under different circumstances using our findings.

However, our analysis does have some limitations. We had limited computing power hence we could run over 2000 experiments and had to sample data. Moreover, we only use R2 for our analysis in this paper and patterns changed depending on metric that was chosen to compute results. For future research we would recommend running experiments with singular variable transformation and running combinations of variables and transformations. Additionally, running more experiments is possible if computing power is increased. We could also enhance performance by using more Feature Engineering techniques and models to see whether results can be improved by them. Lastly, sensitivity analysis can be conducted for the client to help find out which data columns are more sensitive to changes in demand.

REFERENCES

Gartner (2019). "WISEITECH: Machine Learning Automation Platform "WiseProphet"." Retrieved 3/12/2021, 2021, from https://www.gartner.com/teamsiteanalytics/servePDF?g=/imagesrv/media-products/pdf/wiseitech/Wiseitech-1-5UD0614.pdf.

Heaton, J. (2016). An empirical analysis of feature engineering for predictive modeling. SoutheastCon 2016, IEEE.

Khan, S. S. and A. Ahmad (2004). "Cluster center initialization algorithm for K-means clustering." Pattern recognition letters **25**(11): 1293-1302.

Khurana, U., et al. (2018). Feature engineering for predictive modeling using reinforcement learning. Proceedings of the AAAI Conference on Artificial Intelligence.

Kurama, V. (2020). "Gradient Boosting In Classification: Not a Black Box Anymore!". Retrieved 3/12/2021, 2021, from https://blog.paperspace.com/gradient-boosting-for-classification/.

Press, G. (2016) Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says.

Rawat, T. and V. Khemchandani (2017). "Feature engineering (FE) tools and techniques for better classification performance." International Journal of Innovations in Engineering and Technology (IJIET) **8**(2).

Sheriff, S. (2020) The What Why & How of Feature Engineering.

Waqas Khan, P., et al. (2020). "Machine learning based hybrid system for imputation and efficient energy demand forecasting." Energies **13**(11): 2681.

Xie, Y. (2020) Advanced Feature Engineering Could Help Put An End To Rule Complexity Explosion.

Zhang, C., et al. (2018). "On the feature engineering of building energy data mining." Sustainable cities and society **39**: 508-518.

APPENDIX

Table 1. SKU-Level Data for Machine Learning Models

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Tr	Median R2 Te
GB/RF	Count/Mean	None	None	None	None	0.9818	0.9584
				Z-Score Scaling	None	0.9822	0.9620
					PCA	0.9726	0.9297
			Gaussian	None	None	0.9420	0.6730
				Z-Score Scaling	None	0.9460	0.6796
					PCA	0.9119	0.4674
			IQR	None	None	0.6479	0.1916
				Z-Score Scaling	None	0.6486	0.1916
					PCA	0.6725	0.3660
			Quantile	None	None	0.8710	0.2367
				Z-Score Scaling	None	0.8713	0.2367
					PCA	0.9056	0.3059
		Log	None	None	None	0.9818	0.9399
				Z-Score Scaling	None	0.9818	0.9394
					PCA	0.9547	0.7756
			Gaussian	None	None	0.9446	0.6598
				Z-Score Scaling	None	0.9439	0.6564
					PCA	0.9379	0.4447
			IQR	None	None	0.9653	0.5478
				Z-Score Scaling	None	0.9652	0.5442
					PCA	0.9553	0.6428
			Quantile	None	None	0.8712	0.2294
				Z-Score Scaling	None	0.8694	0.2297
					PCA	0.8932	0.2049
		Ratio	None	None	None	0.9763	0.9486
				Z-Score Scaling	None	0.9758	0.9505
					PCA	0.9233	0.2559
			Gaussian	None	None	0.9664	0.8045
				Z-Score Scaling	None	0.9687	0.7652
					PCA	0.9036	0.0501
			IQR	None	None	0.9681	0.7673
				Z-Score Scaling	None	0.9670	0.7464
					PCA	0.9133	0.0863
			Quantile	None	None	0.9684	0.7529
				Z-Score Scaling	None	0.9683	0.7588
					PCA	0.9278	0.1200
		YeoJohnson	None	None	None	0.9809	0.9299
				Z-Score Scaling	None	0.9800	0.9243
					PCA	0.9281	0.6203
			Gaussian	None	None	0.9785	0.9243
				Z-Score Scaling	None	0.9795	0.9189
					PCA	0.9472	0.2201
			IQR	None	None	0.9785	0.9008
				Z-Score Scaling	None	0.9789	0.9038
					PCA	0.9058	0.4769
			Quantile	None	None	0.8712	0.2293
				Z-Score Scaling	None	0.8695	0.2314
					PCA	0.8969	0.3389
	One-Hot	None	None	None	None	0.9952	0.9309
				Z-Score Scaling	None	0.9952	0.9303
					PCA	0.9697	0.9049
			Gaussian	None	None	0.9814	0.6096
				Z-Score Scaling	None	0.9814	0.6138
					PCA	0.9464	0.5961
			IQR	None	None	0.6260	0.1135
				Z-Score Scaling	None	0.6260	0.1134
					PCA	0.6784	0.3460
			Quantile	None	None	0.8363	0.0841
				Z-Score Scaling	None	0.8363	0.0839
					PCA	0.9180	0.4897
		Log	None	None	None	0.9952	0.9108
				Z-Score Scaling	None	0.9952	0.9128
					PCA	0.9563	0.7755
			Gaussian	None	None	0.9662	0.6257
				Z-Score Scaling	None	0.9662	0.6259
					PCA	0.9356	0.4705

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Tr	Median R2 Te
GB/RF	One-Hot	Log	IQR	None	None	0.9928	0.5694
				Z-Score Scaling	None	0.9928	0.5695
					PCA	0.9518	0.7389
			Quantile	None	None	0.8326	0.1320
				Z-Score Scaling	None	0.8326	0.1319
					PCA	0.9175	0.5239
		Ratio	None	None	None	0.9793	0.9533
				Z-Score Scaling	None	0.9813	0.9603
					PCA	0.9276	0.4473
			Gaussian	None	None	0.9672	0.7750
				Z-Score Scaling	None	0.9676	0.7638
					PCA	0.9548	0.1499
			IQR	None	None	0.9677	0.7730
				Z-Score Scaling	None	0.9665	0.7591
			Quantile	None	None	0.9680	0.7549
				Z-Score Scaling	None	0.9676	0.7457
		YeoJohnson	None	None	None	0.9794	0.9362
				Z-Score Scaling	None	0.9799	0.9376
					PCA	0.9298	0.6308
			Gaussian	None	None	0.9777	0.9312
				Z-Score Scaling	None	0.9793	0.9288
					PCA	0.9219	0.4173
			IQR	None	None	0.9781	0.9349
				Z-Score Scaling	None	0.9776	0.9369
					PCA	0.9111	0.2781
			Quantile	None	None	0.8598	0.3050
				Z-Score Scaling	None	0.8604	0.3040
					PCA	0.9031	0.3035

Table 2. SKU-Level Data for Regression Models

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Tr	Median R2 Te
Linear Regression	Count/Mean	None	None	None	None	0.9847	0.9928
				Z-Score Scaling	None	0.9847	0.9928
					PCA	0.9433	0.9378
			Gaussian	None	None	0.3175	0.3598
				Z-Score Scaling	None	0.3175	0.3598
					PCA	0.3072	0.3527
			IQR	Z-Score Scaling	PCA	0.0810	0.0796
			Quantile	None	None	0.1003	0.1005
				Z-Score Scaling	None	0.1003	0.1005
					PCA	0.0932	0.0930
		Log	None	None	None	0.4956	0.5366
				Z-Score Scaling	None	0.4956	0.5366
					PCA	0.4862	0.5255
			Gaussian	None	None	0.1263	0.1358
				Z-Score Scaling	None	0.1263	0.1358
					PCA	0.1212	0.1297
			IQR	Z-Score Scaling	PCA	0.0962	0.0973
			Quantile	None	None	0.0905	0.0904
				Z-Score Scaling	None	0.0905	0.0904
					PCA	0.0874	0.0872
		Ratio	None	None	None	0.9892	0.9811
				Z-Score Scaling	None	0.9892	0.9811
					PCA	0.9486	0.4196
		YeoJohnson	None	None	None	0.2092	0.2435
				Z-Score Scaling	None	0.2092	0.2435
					PCA	0.2050	0.2399
			Gaussian	None	None	0.0888	0.0894
				Z-Score Scaling	None	0.0888	0.0894
					PCA	0.0862	0.0864
			IQR	Z-Score Scaling	PCA	0.0809	0.0793
			Quantile	None	None	0.0873	0.0867
				Z-Score Scaling	None	0.0873	0.0867
					PCA	0.0852	0.0844
	One-Hot	None	None	None	None	0.9849	0.9929
				Z-Score Scaling	None	0.9849	0.9929
					PCA	0.8879	0.9338
			Gaussian	None	None	0.3232	0.3676
				Z-Score Scaling	None	0.3232	0.3676
					PCA	0.3103	0.3567
			IQR	Z-Score Scaling	PCA	0.1065	0.1114
			Quantile	None	None	0.1353	0.1410
				Z-Score Scaling	None	0.1353	0.1410
					PCA	0.1148	0.1205
		Log	None	None	None	0.5163	0.5556
				Z-Score Scaling	None	0.5163	0.5556
					PCA	0.4905	0.5264
			Gaussian	None	None	0.1572	0.1713
				Z-Score Scaling	None	0.1572	0.1713
					PCA	0.1379	0.1507
			IQR	Z-Score Scaling	PCA	0.1142	0.1194
			Quantile	None	None	0.1280	0.1334
				Z-Score Scaling	None	0.1280	0.1334
					PCA	0.1109	0.1169
		Ratio	None	None	None	0.9893	0.9812
				Z-Score Scaling	None	0.9893	0.9812
					PCA	0.9439	0.3163
		YeoJohnson	None	None	None	0.2488	0.2829
				Z-Score Scaling	None	0.2488	0.2829
					PCA	0.2290	0.2661
			Gaussian	None	None	0.1241	0.1299
				Z-Score Scaling	None	0.1241	0.1299
					PCA	0.1097	0.1159
			IQR	Z-Score Scaling	PCA	0.1068	0.1116
			Quantile	None	None	0.1222	0.1267
				Z-Score Scaling	None	0.1222	0.1267

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Tr	Median R2 Te
Regression	One-Hot	YeoJohnson	Quantile	Z-Score Scaling	PCA	0.1091	0.1144
Tweedie/ Huber	Count/Mean	None	None	None	None	**0.9833**	**0.9930**
				Z-Score Scaling	None	**0.7995**	**0.8726**
					PCA	**0.8528**	**0.9206**
			Gaussian	None	None	0.2685	0.2900
				Z-Score Scaling	None	0.2528	0.2618
					PCA	0.2260	0.2321
			IQR	Z-Score Scaling	PCA	0.0713	0.0690
			Quantile	None	None	0.0393	0.0363
				Z-Score Scaling	None	0.0370	0.0348
					PCA	0.0364	0.0343
		Log	None	None	None	0.2103	0.2262
				Z-Score Scaling	None	0.2388	0.2467
					PCA	0.2075	0.2139
			Gaussian	None	None	0.0898	0.0908
				Z-Score Scaling	None	0.0674	0.0652
					PCA	0.0606	0.0582
			IQR	Z-Score Scaling	PCA	0.0664	0.0619
			Quantile	None	None	0.0821	0.0808
				Z-Score Scaling	None	0.0317	0.0291
					PCA	0.0314	0.0288
		Ratio	None	None	None	0.3083	0.2212
				Z-Score Scaling	None	**0.7248**	**0.3455**
					PCA	0.7457	0.2030
			Gaussian	None	None	0.3065	0.2369
				Z-Score Scaling	None	0.3915	0.2275
					PCA	0.3606	0.1955
			Quantile	None	None	0.3086	0.2275
		YeoJohnson	None	None	None	0.1936	0.2260
				Z-Score Scaling	None	0.1638	0.1760
					PCA	0.1627	0.1751
			Gaussian	None	None	0.0808	0.0794
				Z-Score Scaling	None	0.0770	0.0753
					PCA	0.0764	0.0747
			IQR	Z-Score Scaling	PCA	0.0699	0.0672
			Quantile	None	None	0.0797	0.0779
				Z-Score Scaling	None	0.0293	0.0263
					PCA	0.0288	0.0258
	One-Hot	None	None	None	None	**0.9818**	**0.9918**
				Z-Score Scaling	None	**0.6467**	**0.7027**
					PCA	**0.7892**	**0.8531**
			Gaussian	None	None	0.2733	0.2962
				Z-Score Scaling	None	0.2439	0.2524
					PCA	0.2086	0.2111
			IQR	Z-Score Scaling	PCA	0.0953	0.0970
			Quantile	None	None	0.0370	0.0326
				Z-Score Scaling	None	0.0579	0.0586
					PCA	0.0547	0.0557
		Log	None	None	None	0.2260	0.2406
				Z-Score Scaling	None	0.2426	0.2513
					PCA	0.2065	0.2131
			Gaussian	None	None	0.0706	0.0638
				Z-Score Scaling	None	0.0829	0.0830
					PCA	0.0678	0.0687
			IQR	Z-Score Scaling	PCA	0.0523	0.0529
			Quantile	None	None	0.0295	0.0245
				Z-Score Scaling	None	0.0539	0.0545
					PCA	0.0512	0.0520
		Ratio	None	None	None	0.3128	0.2203
				Z-Score Scaling	None	**0.7159**	**0.3549**
					PCA	0.7041	0.2255
			Gaussian	None	None	0.3102	0.2362
				Z-Score Scaling	None	0.3890	0.2328
					PCA	0.3387	0.2116
			Quantile	None	None	0.3062	0.2348

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Tr	Median R2 Te
Tweedie/ Huber	One-Hot	YeoJohnson	None	None	None	0.1596	0.1862
				Z-Score Scaling	None	0.1908	0.2052
					PCA	0.1856	0.2006
			Gaussian	None	None	0.0522	0.0423
				Z-Score Scaling	None	0.0536	0.0527
					PCA	0.0988	0.1010
			IQR	Z-Score Scaling	PCA	0.0951	0.0966
			Quantile	None	None	0.0505	0.0409
				Z-Score Scaling	None	0.0522	0.0526
					PCA	0.0497	0.0503

Table 3. Store By SKU Level Data for Continuous Demand for Machine Learning Models

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Te	Median R2 Tr
Gradient Boosting	Count/Mean	None	None	None	None	0.7337	0.8722
				Z-Score	PCA	0.7278	0.8562
			Gaussian	None	None	0.6603	0.6628
				Z-Score	PCA	0.5240	0.6390
			IQR	None	None	0.4842	0.4917
				Z-Score	PCA	0.4309	0.4856
			Quantile	None	None	0.5073	0.5158
				Z-Score	PCA	0.4362	0.4946
		Log	None	None	None	0.7701	0.8660
				Z-Score	PCA	0.5949	0.7396
			Gaussian	None	None	0.7385	0.8118
				Z-Score	PCA	0.5571	0.6463
			IQR	None	None	0.7406	0.8473
				Z-Score	PCA	0.7165	0.7938
			Quantile	None	None	0.7350	0.8621
				Z-Score	PCA	0.3898	0.5002
		Ratio	None	None	None	0.8710	0.8810
				Z-Score	PCA	0.7179	0.8715
			Gaussian	None	None	0.8819	0.9357
				Z-Score	PCA	0.7252	0.8754
			IQR	None	None	0.8660	0.9353
				Z-Score	PCA	0.7317	0.8742
			Quantile	None	None	0.8703	0.9359
				Z-Score	PCA	0.7343	0.8743
		YeoJohnson	None	None	None	0.7018	0.7668
				Z-Score	PCA	0.5812	0.7234
			Gaussian	None	None	0.7408	0.8568
				Z-Score	PCA	0.5680	0.6662
			IQR	None	None	0.7122	0.8541
				Z-Score	PCA	0.6675	0.7281
			Quantile	None	None	0.6612	0.6646
				Z-Score	PCA	0.4140	0.4772
	One-Hot	None	None	None	None	0.5732	0.6174
				Z-Score	PCA	0.7224	0.8569
			Gaussian	None	None	0.5579	0.5383
				Z-Score	PCA	0.5369	0.6442
			IQR	None	None	0.4727	0.4570
				Z-Score	PCA	0.4319	0.5043
			Quantile	None	None	0.4831	0.4647
				Z-Score	PCA	0.4305	0.5200
		Log	None	None	None	0.7009	0.7708
				Z-Score	PCA	0.5960	0.7496
			Gaussian	None	None	0.7402	0.7671
				Z-Score	PCA	0.5263	0.6446
			IQR	None	None	0.7403	0.8466
				Z-Score	PCA	0.7105	0.7939
			Quantile	None	None	0.6982	0.7686
				Z-Score	PCA	0.4227	0.4944
		Ratio	None	None	None	0.8939	0.9257
				Z-Score	PCA	0.7366	0.8724
			Gaussian	None	None	0.8628	0.9310
				Z-Score	PCA	0.7366	0.8725
			IQR	None	None	0.8645	0.9314
				Z-Score	PCA	0.7345	0.8738

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Te	Median R2 Tr
Gradient Boosting	**One-Hot**	**Ratio**	**Quantile**	**None**	**None**	**0.8943**	**0.9314**
				Z-Score	**PCA**	**0.7427**	**0.8737**
		YeoJohnson	**None**	**None**	**None**	**0.7419**	**0.8677**
				Z-Score	**PCA**	**0.6182**	**0.7064**
			Gaussian	**None**	**None**	**0.8020**	**0.8567**
				Z-Score	**PCA**	**0.5704**	**0.6630**
			IQR	**None**	**None**	**0.7142**	**0.8556**
				Z-Score	**PCA**	**0.6719**	**0.7438**
			Quantile	**None**	**None**	**0.5313**	**0.5436**
				Z-Score	**PCA**	**0.4167**	**0.4739**

Table 4. Store By SKU Level Data for Continuous Demand for Regression Models

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Te	Median R2 Tr
Huber/ Tweedie Regression	Count/Mean	None	None	None	None	0.1485	0.1994
				Z-Score	None	0.2808	0.3819
			Gaussian	None	None	0.1548	0.2141
				Z-Score	None	0.2512	0.3541
			IQR	None	None	0.1417	0.1982
				Z-Score	None	0.2323	0.3279
			Quantile	None	None	0.1489	0.2076
				Z-Score	None	0.2355	0.3281
		Log	None	None	None	0.1711	0.2230
				Z-Score	None	0.2729	0.3705
			Gaussian	None	None	0.1786	0.2463
				Z-Score	None	0.2482	0.3457
			IQR	None	None	0.1629	0.2270
				Z-Score	None	0.2345	0.3273
			Quantile	None	None	0.1745	0.2416
				Z-Score	None	0.2339	0.3252
		Ratio	None	None	None	0.1208	0.1737
				Z-Score	None	0.0451	0.4337
			Gaussian	None	None	0.1366	0.2269
				Z-Score	None	0.1266	0.4385
			IQR	None	None	0.1615	0.2274
				Z-Score	None	0.2221	0.4380
			Quantile	None	None	0.1428	0.2271
				Z-Score	None	0.2037	0.4376
		YeoJohnson	None	None	None	0.1112	0.1588
				Z-Score	None	0.2130	0.3030
			Gaussian	None	None	0.1131	0.1621
				Z-Score	None	0.2117	0.3010
			IQR	None	None	0.1122	0.1605
				Z-Score	None	0.2113	0.3006
			Quantile	None	None	0.1316	0.1882
				Z-Score	None	0.2070	0.2942
	One-Hot	None	None	None	None	0.1488	0.1998
				Z-Score	None	0.2888	0.3939
			Gaussian	None	None	0.1549	0.2143
				Z-Score	None	0.2620	0.3697
			IQR	None	None	0.1412	0.1975
				Z-Score	None	0.2468	0.3488
			Quantile	None	None	0.1499	0.2091
				Z-Score	None	0.2487	0.3471
		Log	None	None	None	0.1829	0.2326
				Z-Score	None	0.2829	0.3858
			Gaussian	None	None	0.1904	0.2641
				Z-Score	None	0.2603	0.3631
			IQR	None	None	0.1889	0.2629
				Z-Score	None	0.2480	0.3468
			Quantile	None	None	0.1734	0.2375
				Z-Score	None	0.2468	0.3439
		Ratio	None	None	None	0.1239	0.1718
				Z-Score	None	0.0520	0.4325
			Gaussian	None	None	0.1439	0.2270
				Z-Score	None	0.1242	0.4449
			IQR	None	None	0.1242	0.2265
				Z-Score	None	0.2152	0.4448

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Te	Median R2 Tr
Huber/ Tweedie Regression	One-Hot	Ratio	Quantile	None	None	0.1451	0.2304
				Z-Score	None	0.1955	0.4442
		YeoJohnson	None	None	None	0.1288	0.1828
				Z-Score	None	0.2322	0.3300
			Gaussian	None	None	0.1304	0.1862
				Z-Score	None	0.2309	0.3281
			IQR	None	None	0.1354	0.1928
				Z-Score	None	0.2303	0.3274
			Quantile	None	None	0.1293	0.1839
				Z-Score	None	0.2264	0.3214
Linear Regression	Count/Mean	None	None	None	None	**0.7084**	**0.8293**
				Z-Score	PCA	**0.7056**	**0.8242**
			Gaussian	None	None	**0.5764**	**0.5230**
				Z-Score	PCA	0.4840	0.4405
			IQR	None	None	0.3662	0.3244
				Z-Score	PCA	0.3514	0.3110
			Quantile	None	None	0.3988	0.3549
				Z-Score	PCA	0.3737	0.3314
		Log	None	None	None	0.4333	0.3881
				Z-Score	PCA	0.4499	0.4052
			Gaussian	None	None	0.4060	0.3607
				Z-Score	PCA	0.4004	0.3552
			IQR	None	None	0.3929	0.3511
				Z-Score	PCA	0.3808	0.3406
			Quantile	None	None	0.3559	0.3148
				Z-Score	PCA	0.3507	0.3103
		Ratio	None	None	None	0.3997	0.3562
				Z-Score	PCA	**0.7800**	**0.8889**
			Gaussian	None	None	0.3784	0.3357
				Z-Score	PCA	**0.7775**	**0.8858**
			IQR	None	None	0.3854	0.3415
				Z-Score	PCA	**0.7803**	**0.8878**
			Quantile	None	None	0.4359	0.3892
				Z-Score	PCA	**0.7746**	**0.8882**
		YeoJohnson	None	None	None	0.4085	0.3637
				Z-Score	PCA	0.3813	0.3402
			Gaussian	None	None	0.4137	0.3679
				Z-Score	PCA	0.3755	0.3332
			IQR	None	None	0.3988	0.3545
				Z-Score	PCA	0.3407	0.3020
			Quantile	None	None	0.3914	0.3483
				Z-Score	PCA	0.3297	0.2912
	One-Hot	None	None	None	None	**0.7083**	**0.8294**
				Z-Score	PCA	**0.6993**	**0.8159**
			Gaussian	None	None	**0.5778**	**0.5238**
				Z-Score	PCA	0.4888	0.4434
			IQR	None	None	0.3691	0.3261
				Z-Score	PCA	0.3556	0.3128
			Quantile	None	None	0.4031	0.3576
				Z-Score	PCA	0.3843	0.3391
		Log	None	None	None	0.4220	0.3766
				Z-Score	PCA	0.4541	0.4084
			Gaussian	None	None	0.4139	0.3661
				Z-Score	PCA	0.4123	0.3645

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Te	Median R2 Tr
Linear Regression	One-Hot	Log	IQR	None	None	0.3976	0.3544
				Z-Score	PCA	0.3888	0.3463
			Quantile	None	None	0.4320	0.3855
				Z-Score	PCA	0.3602	0.3169
		Ratio	None	None	None	0.4322	0.3847
				Z-Score	PCA	**0.7825**	**0.8904**
			Gaussian	None	None	0.4063	0.3622
				Z-Score	PCA	**0.7789**	**0.8853**
			IQR	None	None	0.3826	0.3389
				Z-Score	PCA	**0.7831**	**0.8881**
			Quantile	None	None	0.3666	0.3252
				Z-Score	PCA	**0.7729**	**0.8869**
		YeoJohnson	None	None	None	0.4075	0.3618
				Z-Score	PCA	0.3846	0.3415
			Gaussian	None	None	0.5046	0.4538
				Z-Score	PCA	0.3800	0.3356
			IQR	None	None	0.3689	0.3263
				Z-Score	PCA	0.3485	0.3077
			Quantile	None	None	0.3861	0.3419
				Z-Score	PCA	0.3373	0.2963
Ridge Regression	Count/Mean	None	None	None	None	**0.5537**	**0.5915**
				Z-Score	PCA	**0.7056**	**0.8242**
			Gaussian	None	None	**0.5754**	**0.5223**
				Z-Score	PCA	0.4840	0.4405
			IQR	None	None	0.3664	0.3248
				Z-Score	PCA	0.3514	0.3110
			Quantile	None	None	0.4086	0.3637
				Z-Score	PCA	0.3737	0.3314
		Log	None	None	None	0.4287	0.3843
				Z-Score	PCA	0.4499	0.4052
			Gaussian	None	None	0.4060	0.3607
				Z-Score	PCA	0.4004	0.3552
			IQR	None	None	0.3930	0.3511
				Z-Score	PCA	0.3808	0.3406
			Quantile	None	None	0.4139	0.3661
				Z-Score	PCA	0.3507	0.3103
		Ratio	None	None	None	0.3556	0.3148
				Z-Score	PCA	**0.7800**	**0.8889**
			Gaussian	None	None	0.4559	0.4108
				Z-Score	PCA	**0.7775**	**0.8858**
			IQR	None	None	0.3933	0.3514
				Z-Score	PCA	**0.7803**	**0.8878**
			Quantile	None	None	0.3926	0.3509
				Z-Score	PCA	**0.7746**	**0.8882**
		YeoJohnson	None	None	None	0.4078	0.3626
				Z-Score	PCA	0.3813	0.3402
			Gaussian	None	None	0.3963	0.3526
				Z-Score	PCA	0.3755	0.3332
			IQR	None	None	0.3753	0.3349
				Z-Score	PCA	0.3407	0.3020
			Quantile	None	None	0.3559	0.3162
				Z-Score	PCA	0.3297	0.2912
	One-Hot	None	None	None	None	0.4047	0.3576
				Z-Score	PCA	**0.6993**	**0.8159**

Model	Categorical	Numeric	Outlier	Scaler	PCA	Median R2 Te	Median R2 Tr
Ridge Regression	One-Hot	None	Gaussian	None	None	0.5778	0.5238
				Z-Score	PCA	0.4888	0.4434
			IQR	None	None	0.3692	0.3261
				Z-Score	PCA	0.3556	0.3128
			Quantile	None	None	0.4031	0.3576
				Z-Score	PCA	0.3843	0.3391
		Log	None	None	None	0.4451	0.3985
				Z-Score	PCA	0.4541	0.4084
			Gaussian	None	None	0.4139	0.3661
				Z-Score	PCA	0.4123	0.3645
			IQR	None	None	0.3963	0.3538
				Z-Score	PCA	0.3888	0.3463
			Quantile	None	None	0.3647	0.3233
				Z-Score	PCA	0.3602	0.3169
		Ratio	None	None	None	0.4544	0.4101
				Z-Score	PCA	0.7825	0.8904
			Gaussian	None	None	0.3846	0.3450
				Z-Score	PCA	0.7789	0.8853
			IQR	None	None	0.4555	0.4109
				Z-Score	PCA	0.7831	0.8881
			Quantile	None	None	0.3463	0.3073
				Z-Score	PCA	0.7729	0.8869
		YeoJohnson	None	None	None	0.4122	0.3659
				Z-Score	PCA	0.3846	0.3415
			Gaussian	None	None	0.4047	0.3575
				Z-Score	PCA	0.3800	0.3356
			IQR	None	None	0.3961	0.3536
				Z-Score	PCA	0.3485	0.3077
			Quantile	None	None	0.4286	0.3808
				Z-Score	PCA	0.3373	0.2963

Table 5. Data For Machine Learning Models

Model	Categorical	Numeric	Scaler	Outlier	Median R2 Tr	Median R2 Te
Gradient Boosting	Count/Mean	None	None	None	0.5253	0.37
				Gaussian	0.5171	0.36
				IQR	0.5057	0.36
				Quantile	0.5053	0.36
			Z-Score	None	0.5041	0.35
				Gaussian	0.4981	0.34
				IQR	0.4873	0.34
				Quantile	0.4849	0.33
		YeoJohnson	Z-Score	None	0.4643	0.32
				Gaussian	0.4613	0.31
				IQR	0.4624	0.32
				Quantile	0.4443	0.30
	One-Hot	None	None	None	0.5279	0.37
				Gaussian	0.5188	0.36
				IQR	0.5081	0.36
				Quantile	0.5073	0.36
			Z-Score	None	0.5020	0.35
				Gaussian	0.4955	0.34
				IQR	0.4865	0.34
				Quantile	0.4809	0.33
		YeoJohnson	Z-Score	None	0.4623	0.31
				Gaussian	0.4573	0.31
				IQR	0.4599	0.31
				Quantile	0.4358	0.29

Table 6. Data For Regression Models

Model	Categorical	Numeric	Scaler	Outlier	Median R2 Tr	Median R2 Te
Huber/ Tweedie Regression	Count/Mean	None	None	None	0.1994	0.15
				Gaussian	0.2141	0.15
				IQR	0.1982	0.14
				Quantile	0.2076	0.15
			Z-Score	None	**0.3819**	**0.28**
				Gaussian	**0.3541**	**0.25**
				IQR	0.3279	0.23
				Quantile	**0.3281**	**0.24**
		YeoJohnson	None	None	0.1588	0.11
				Gaussian	0.1621	0.11
				IQR	0.1605	0.11
				Quantile	0.1882	0.13
			Z-Score	None	0.3030	0.21
				Gaussian	0.3010	0.21
				IQR	0.3006	0.21
				Quantile	0.2942	0.21
		Log	None	None	0.2230	0.17
				Gaussian	0.2463	0.18
				IQR	0.2270	0.16
				Quantile	0.2416	0.17
			Z-Score	None	**0.3705**	**0.27**
				Gaussian	**0.3457**	**0.25**
				IQR	0.3273	0.23
				Quantile	0.3252	0.23
		Ratio	None	None	0.1737	0.12
				Gaussian	0.2269	0.14
				IQR	0.2274	0.16
				Quantile	0.2271	0.14
			Z-Score	None	0.4337	0.05
				Gaussian	0.4385	0.13
				IQR	0.4380	0.22
				Quantile	0.4376	0.20
	One-Hot	None	None	None	0.1998	0.15
				Gaussian	0.2143	0.15
				IQR	0.1975	0.14
				Quantile	0.2091	0.15
			Z-Score	None	**0.3939**	**0.29**
				Gaussian	**0.3697**	**0.26**
				IQR	**0.3488**	**0.25**
				Quantile	**0.3471**	**0.25**
		YeoJohnson	None	None	0.1828	0.13
				Gaussian	0.1862	0.13
				IQR	0.1928	0.14
				Quantile	0.1839	0.13
			Z-Score	None	0.3300	0.23
				Gaussian	0.3281	0.23
				IQR	0.3274	0.23
				Quantile	0.3214	0.23
		Log	None	None	0.2326	0.18
				Gaussian	0.2641	0.19
				IQR	0.2629	0.19
				Quantile	0.2375	0.17
			Z-Score	None	**0.3858**	**0.28**
				Gaussian	**0.3631**	**0.26**

Model	Categorical	Numeric	Scaler	Outlier	Median R2 Tr	Median R2 Te
Huber/ Tweedie Regression	One-Hot	Log	Z-Score	IQR	**0.3468**	**0.25**
				Quantile	**0.3439**	**0.25**
		Ratio	None	None	0.1718	0.12
				Gaussian	0.2270	0.14
				IQR	0.2265	0.12
				Quantile	0.2304	0.15
			Z-Score	None	0.4325	0.05
				Gaussian	0.4449	0.12
				IQR	0.4448	0.22
				Quantile	0.4442	0.20
Linear Regression	Count/Mean	None	None	None	**0.4331**	**0.32**
				Gaussian	**0.4089**	**0.29**
				IQR	**0.3823**	**0.27**
				Quantile	**0.3831**	**0.28**
			Z-Score	None	**0.4315**	**0.32**
				Gaussian	**0.4071**	**0.29**
				IQR	**0.3804**	**0.27**
				Quantile	**0.3797**	**0.27**
		YeoJohnson	Z-Score	None	**0.3598**	**0.25**
				Gaussian	**0.3568**	**0.25**
				IQR	**0.3555**	**0.25**
				Quantile	**0.3465**	**0.24**
		Log	Z-Score	None	**0.4192**	**0.31**
				Gaussian	**0.3941**	**0.28**
				IQR	**0.3743**	**0.26**
				Quantile	**0.3699**	**0.26**
		Ratio	Z-Score	None	**0.4542**	**0.33**
				Gaussian	**0.4610**	**0.32**
				IQR	0.4582	0.19
				Quantile	**0.4575**	**0.25**
	One-Hot	None	None	None	**0.4434**	**0.32**
				Gaussian	**0.4222**	**0.30**
				IQR	**0.4011**	**0.29**
				Quantile	**0.3987**	**0.29**
			Z-Score	None	**0.4409**	**0.32**
				Gaussian	**0.4199**	**0.30**
				IQR	**0.3981**	**0.28**
				Quantile	**0.3947**	**0.28**
		YeoJohnson	Z-Score	None	**0.3797**	**0.27**
				Gaussian	**0.3768**	**0.27**
				IQR	**0.3754**	**0.26**
				Quantile	**0.3657**	**0.26**
		Log	Z-Score	None	**0.4290**	**0.31**
				Gaussian	**0.4098**	**0.29**
				IQR	**0.3905**	**0.28**
				Quantile	**0.3863**	**0.27**
		Ratio	Z-Score	None	**0.4607**	**0.32**
				Gaussian	**0.4679**	**0.33**
				IQR	**0.4667**	**0.30**
				Quantile	**0.4659**	**0.31**