Antonio Gonzalez

Udacity Machine Learning Engineer Capstone Report

# Project: Applied Machine Learning to Predict Banking Transaction Values

## Introduction

This project is based on a Kaggle competition hosted by Santander in 2018. The competition's goal was to accurately predict the financial value of a customer transaction based on unique attributes of the customer and their past interactions. The project was sponsored by Santander with the broader goal of more accurately anticipating and addressing customer needs in a personalized manner.

Rising consumer expectations for personalized service, coupled with new market entrants that are equipped to cost effectively address those expectations, are some of the most fundamental competitive pressures now facing financial services. An industry that has traditionally relied on the personal touch in deepening relationships has had to rethink the meaning of personalization when many interactions are now digital touchpoints that do not afford the opportunity for human-based interaction. Consequently, many financial institutions are looking to invest heavily in predictive analytics. I decided to try my hand at this competition to better understand the dynamics of consumer needs, as well as the nature of investments firms are making.

## Project Goals

In this project, I am to develop a solution that will be able to accurately predict the value of transactions for each banking customer in a provided test dataset. Predictive outputs will be compared to actual financial data for customers in the test dataset. The evaluation metric will be a comparison of the model outputs to actual results using Root Mean Squared Logarithmic Error (RMSLE). RMSLE is appropriate in this case because of the likely presence of very significant outliers in cases of financial transactions, as well as the greater cost of underpredicting the value of a banking relationship relative to that of overvaluation.

A good baseline score to exceed will be the evaluation metric score on a basic gradient boosting model without any feature selection or transformation - a RMSLE of approximately 1.47.

## The Data

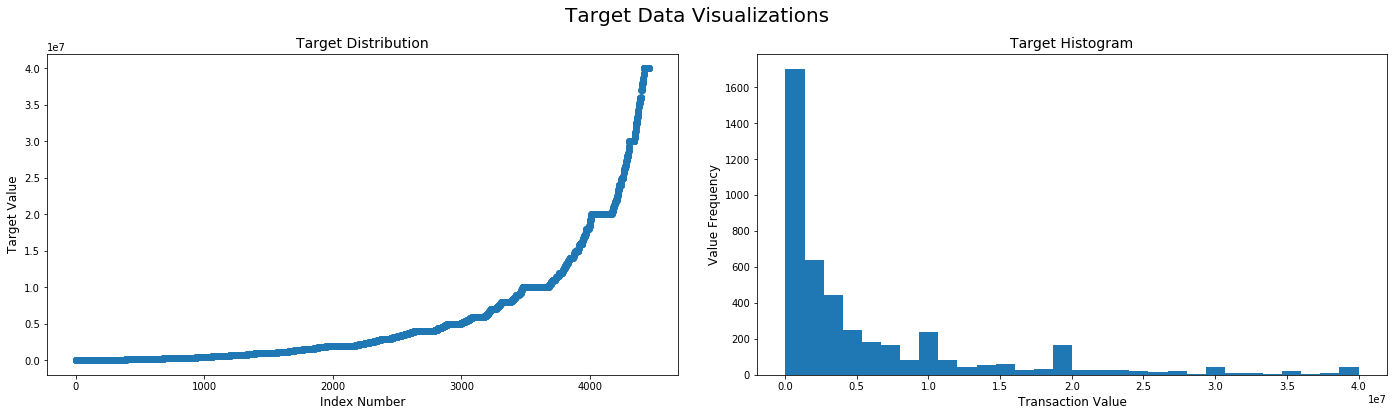
The competition is based on anonymized customer data provided by Santander. Each row of the data is intended to represent an individual customer, with an associated 4,993 columns. The target is a continuous variable of an unknown financial denomination, one column represents a unique ID tag, and the remaining 4,991 columns represent anonymized features for which no information is provided. Consequently, domain knowledge of financial services will be unavailable in solving this problem, presenting an interesting opportunity to solve a problem using only the statistical properties of the features.

Two datasets are provided with the characteristics described above. The first, a training dataset, includes 4,459 rows. Most feature values are zero, with a small number of large outliers, indicating that significant data processing will be needed to deal with these significantly left skewed distributions. In addition, the high ratio of features to observations will necessitate feature reduction. The second dataset, a test set, includes 49,342 rows.

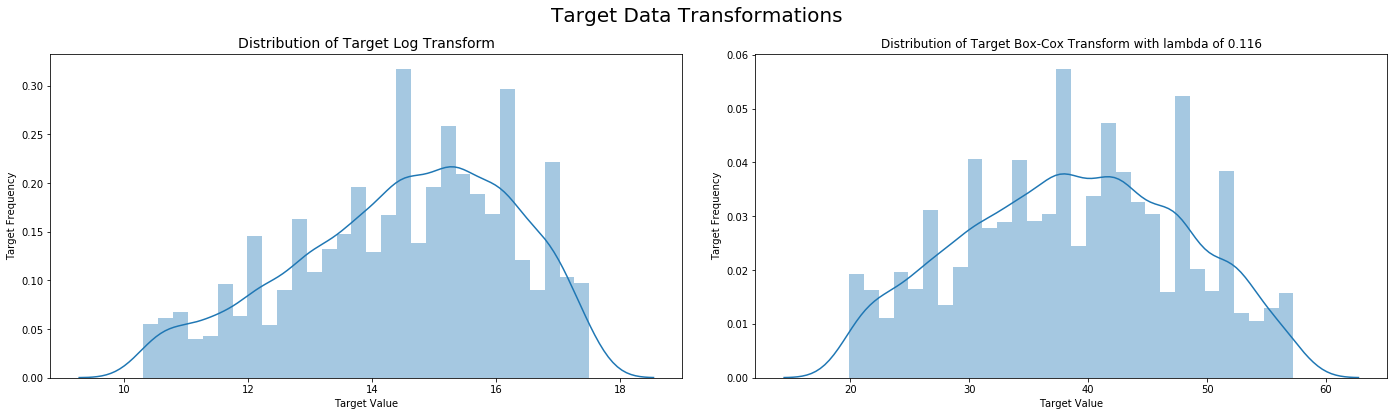
Based on my exploration of the competition discussions, it appears a leak was discovered early on and reported to the competition host. It was discovered that some rows do not actually represent unique individuals but instead time series values of the same customers. Ultimately, Santander decided that the leak was reflective of their internal data and would therefore be incorporated into their predictive analytics solutions, and so winning solutions exploited the leak to achieve RMSLE scores approaching .5.

## Data Exploration and Visualization

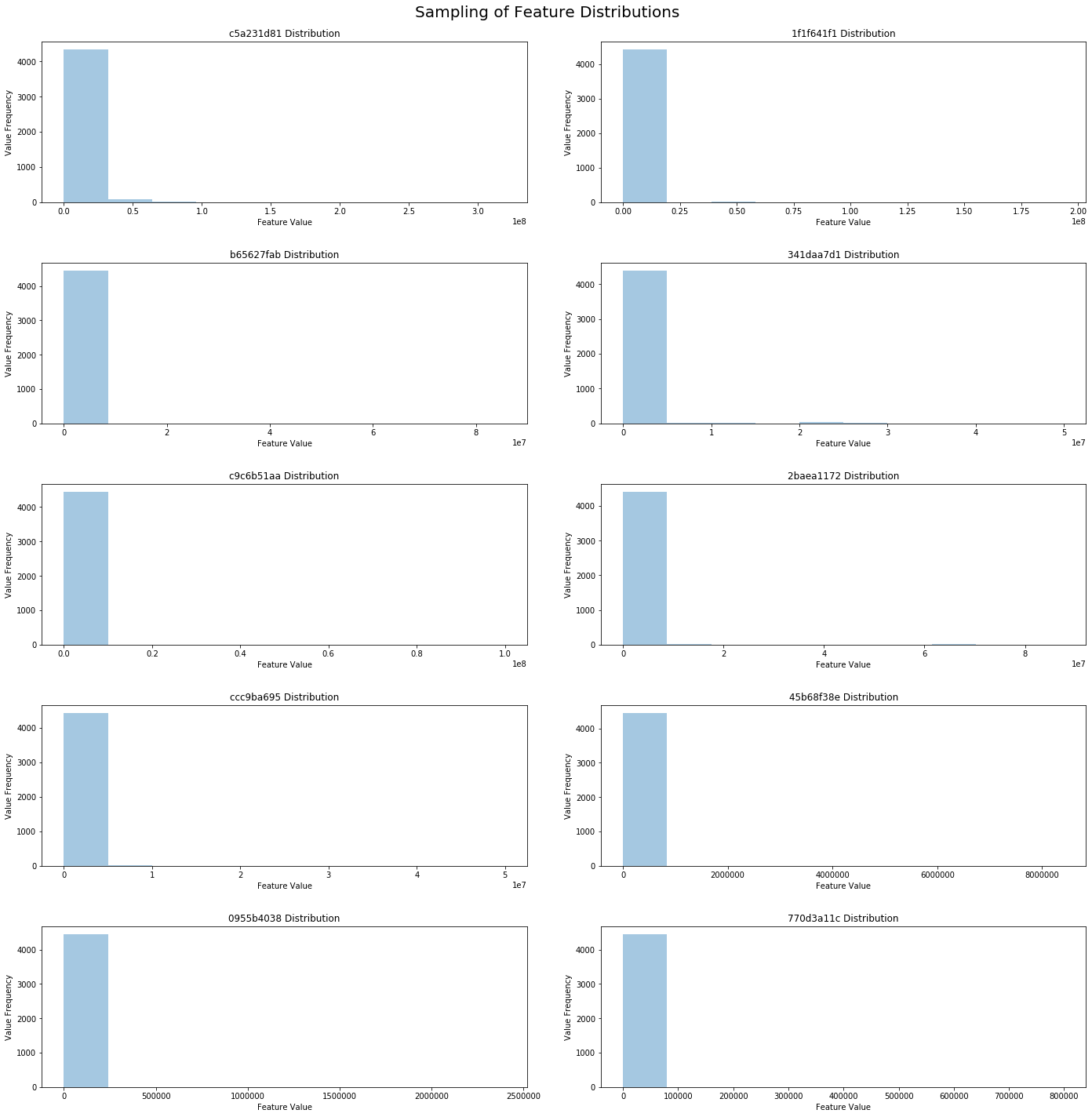
An initial exploration of the data indicates significant processing will be necessary. I begin by examining the target data, a continuous variable with a minimum value of 30000 and a maximum value of 40,000,000. A visualization of the target data distribution shows this strong right skew, with range of significant outliers.



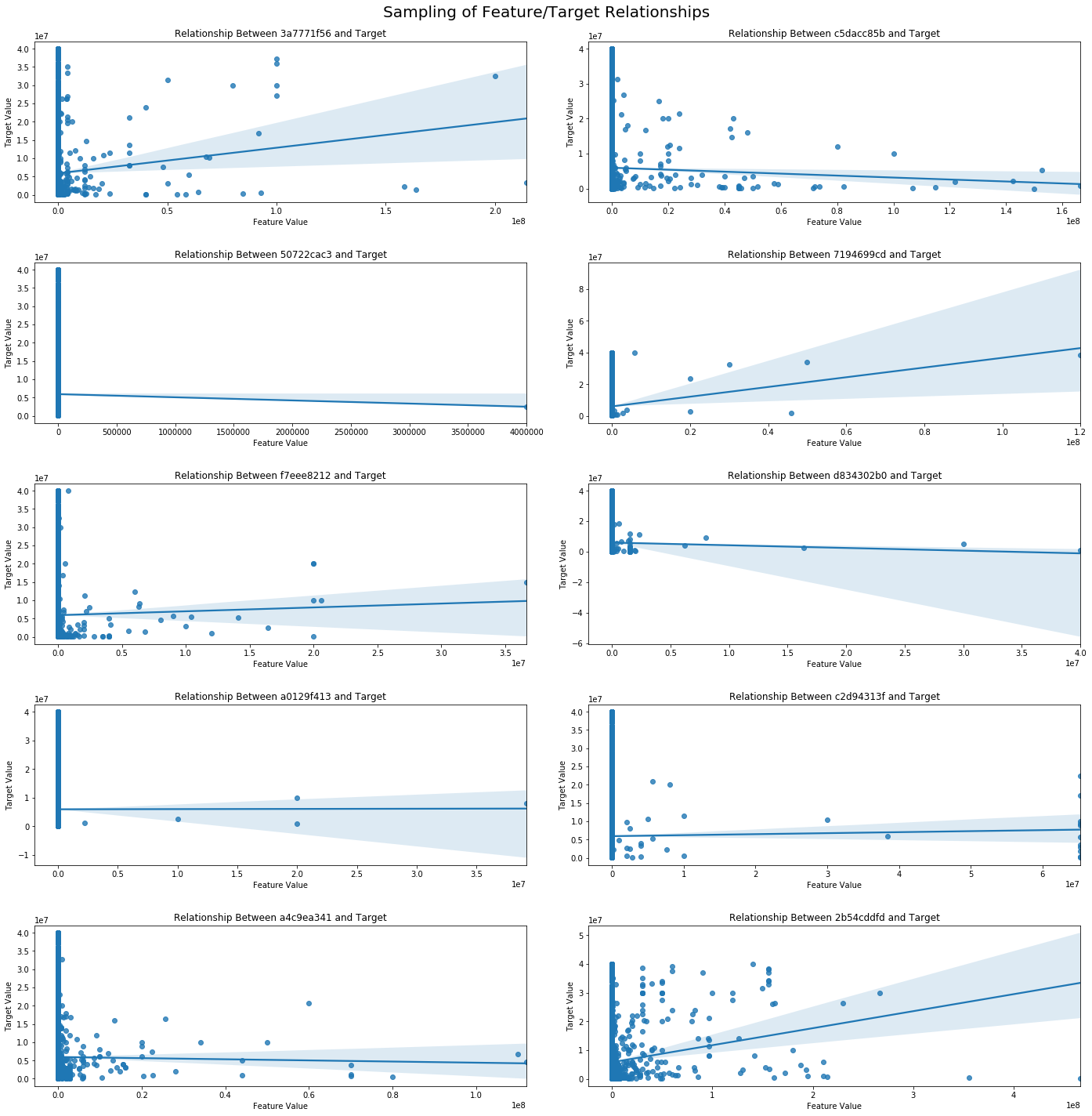
To remedy this, I explored two potential transformations to the target, a log transform and a Box-Cox transform. In the event that the log transform is inadequate to fully address the skew, I included a Box-Cox to find an appropriate lambda maximizes the log-likelihood function. As shown below, a Box-Cox with a lambda of ~.116 achieves the best result.



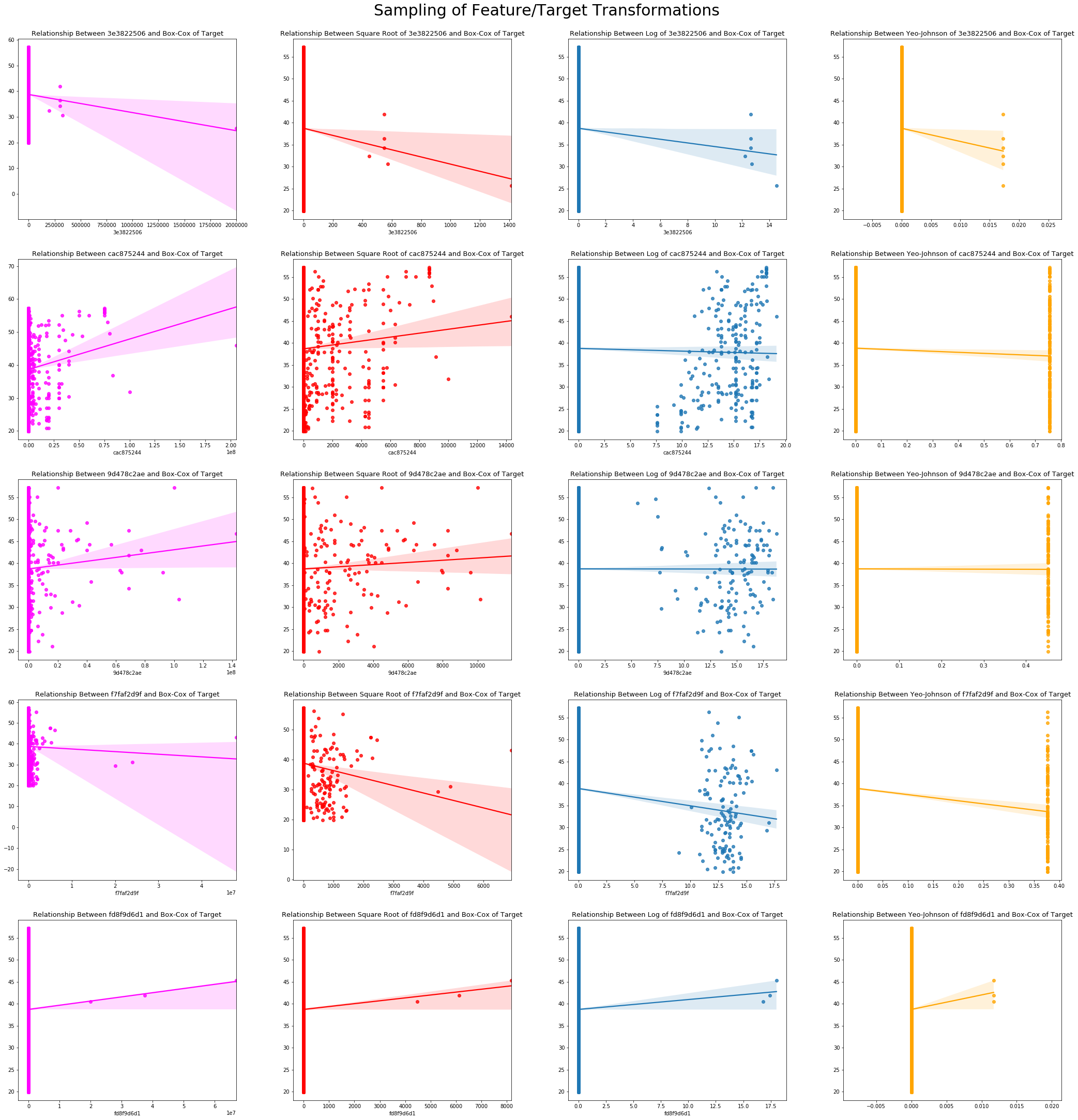
Next, I examined the training data features. Because there are nearly 5,000 features, with no descriptive names on which to infer a model, I conducted an initial evaluation of feature distributions of a sample of 10 features, presented below:



As shown above, a very significant majority of feature values are zero, with larger outliers sporadically present. Consequently, the relationships between features and the target generally have a high degree of bias introduced by a small number of inflated values.



I am not aware of any transformations explicitly designed to address sparsity in feature data, and so examined the performance of three different methods to address right skew in data. First, I simply apply the Box-Cox transform to the target data and examine the relationship to the untransformed features. Second, I took the square root of the feature values and visualized their relationship to the Box-Cox of the target. Third, I again transform the target with a Box-Cox but then apply a log transformation to the features. Because most values are zeros, I add a constant of 1 to the feature data before the transformation. Last, I transformed the target again with a Box-Cox and attempted to conducted a similar optimized function to the feature data. Because of the present of negative values in the features, I applied a Yeo-Johnson power transformation. I visualized the relationships between these transformed data using a smaller sample of five features.

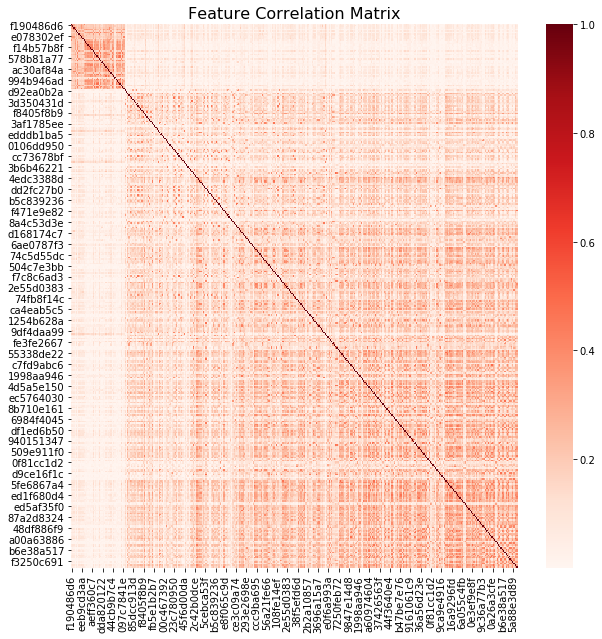


The relationship between log transformed features and the Box-Cox transformed target appears to take on the most definite shape. I calculated the average absolute Spearman’s Rank-Order value for each of the transformations and found all three had similarly low values, potentially indicating a nonlinear relationship between features and target or generally small impact of any given feature.

## First Model: Feature Elimination and Random Forest Regression

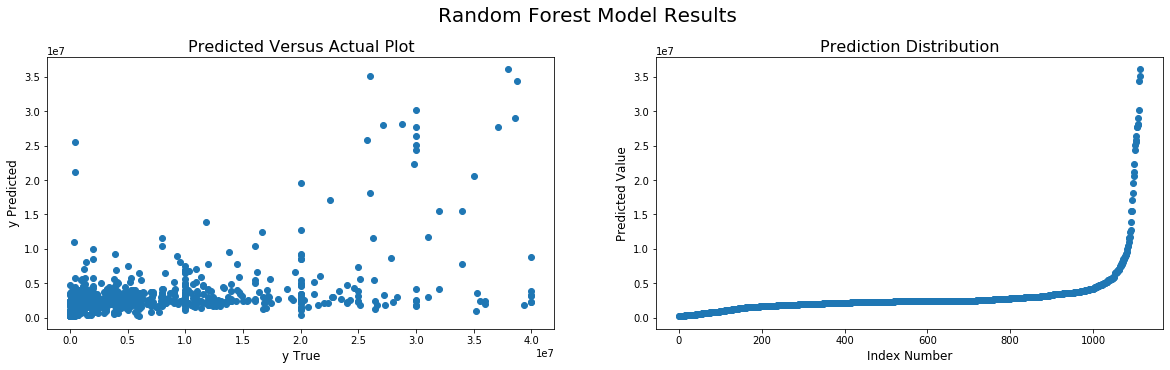
In my first pass at creating a predictive model, I processed and reduced features manually before incorporating them into a model. I began by reducing the number of features, firstly out of computational capacity limitations as well as to reduce model bias. First, I calculated the overall correlation of each feature to the target using Spearman’s Rank-Order. I opted to use a Spearman correlation over Pearson because of the presence of large outliers in the target values, which a rank-order approach could better address. I decided to use features with a minimum correlation coefficient of .05, under which improvements to the model diminished rapidly. Applying this threshold, I further reduced the features down from 4,735 to 585.

Next, I reduced features based on their correlation to one another. I used a threshold of .5, which I found produced the highest model accuracy while effectively reducing the feature count down to 348 features. The resulting correlation heatmap of features appeared as follows:



After reducing the feature count, I conducted the transformations I felt were most appropriate for the data. I applied a Box-Cox transformation to the target values to normalize them prior to any analysis. Because no feature transformation clearly resulted in stronger feature/target relationships, I evaluated the effect of each transformation through trial and error. Ultimately, I found taking the square root of features resulted in the highest preliminary RMSLE of approximately 1.52.

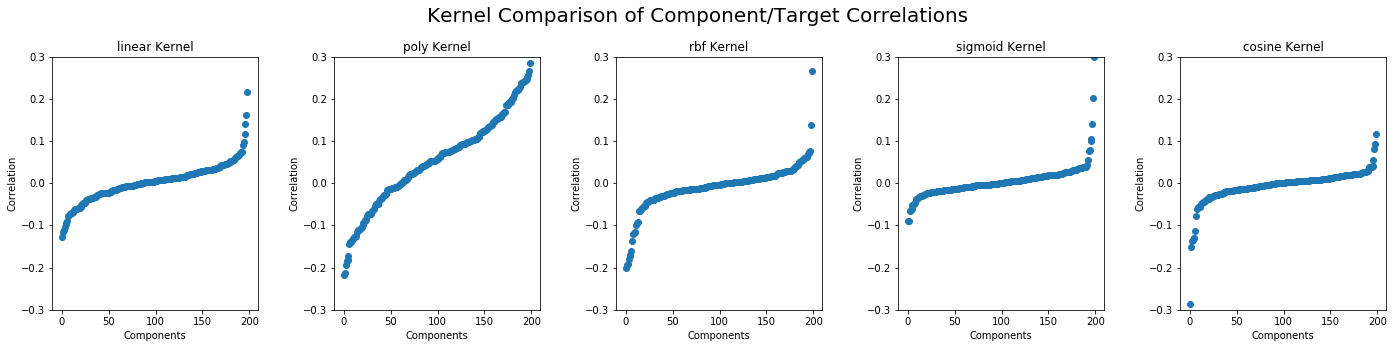
Having processed the features in this way, I next moved to model selection. After conducting a Kfolds comparison of different baseline models, I found the random forest regression performed the best. This makes sense, given the high dimensionality of the data and almost definitely nonlinear relationships between features and the target. However, a preliminary examination of the predictions made on validation data show a persistent underestimation of results. The plateauing of the right chart indicates the same prediction is being made for the majority of observations.



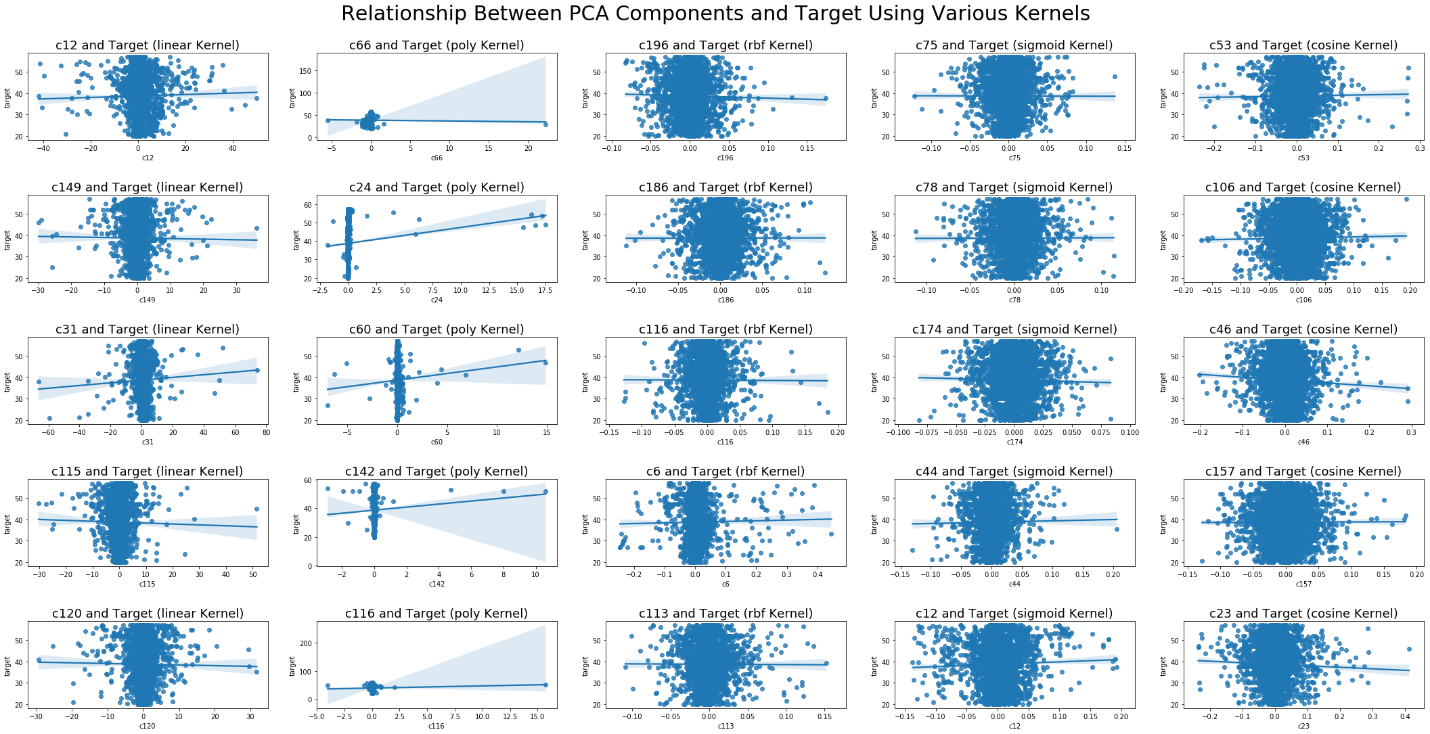
Despite receiving an RSMLE of 1.52 on the split of the training data, predictions made by the model on the test data performed significantly worse, receiving an RSMLE of 1.79. I decided to refine my approach to feature reduction by using a PCA approach, having decided that one of the most significant issues with the model was the significant sparsity of data. Individual observations only had values for a small fraction of features, and so training has the risk of significantly overfitting to the feature values that happen to exist in the train data but not test, and vice versa. PCA could potentially overcome the challenge of sparsity by consolidating values across many features as lower dimensional components.

## Second Model: Principal Component Analysis and Random Forest Regression

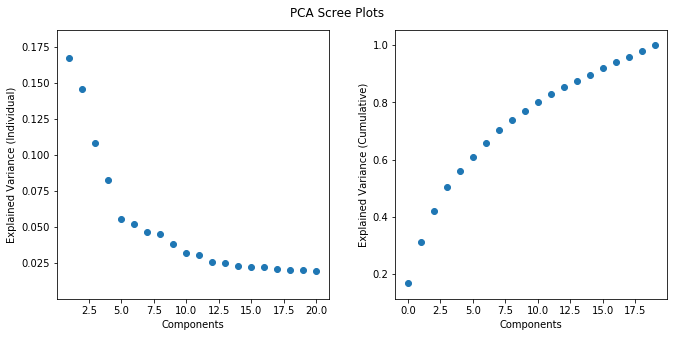
I hypothesized that principal components could be a better approach to feature reduction by consolidating values and reducing the impact of data sparsity. Moreover, the anonymized nature of the variables in this problem means that the meanings of PCA components will at the very least not be any less clear. Rather than assume linear relationships among the features, I opted to cross validate with multiple potential kernels. An initial analysis of component correlations to the target across kernels indicates a polynomial PCA may provide the strongest results.



The polynomial kernel has a larger number of components with a significant correlation to the target values, indicating at least the largest opportunity to reduce the number of components further based on their correlation. A visual analysis of the relationship between components and the target affirms that a linear or polynomial-based PCA will provide the strongest results:

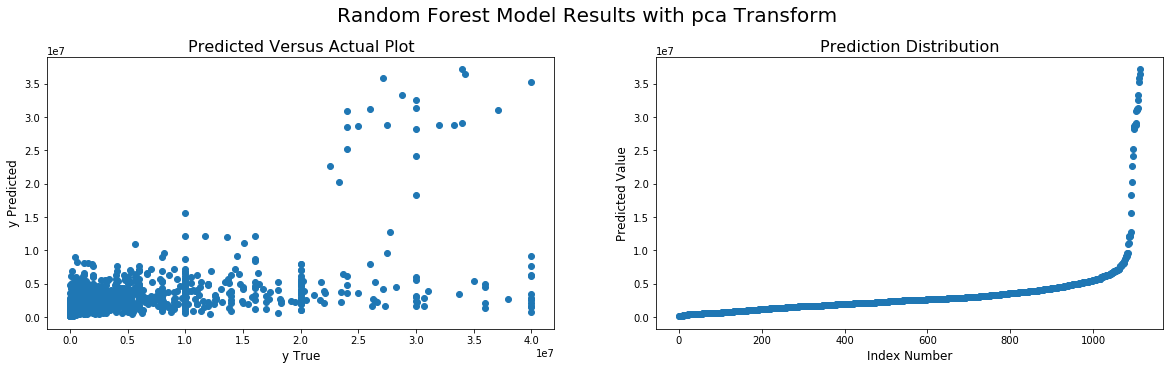


The linear and polynomial kernels appear to be best suited to address outliers by producing components with a wider range of values. Meanwhile, rbf, cosine, and sigmoid kernels appear to be limited to exceedingly small values centered around zero. The linear kernel appears to have a good spread of values without significant outliers, and so I opted to use that in my analysis. After fitting and transforming the data, I visualized the explained variance of the 20 components, both independently and cumulatively.



The scree plots indicate that PCA works moderately well on the dataset, with a defined elbow in explained variance at approximately five components. I reduced the components down to those accounting for 80% of explained variance, resulting in a total of ten components.

Applying a random forest regressor model to the PCA reduced data achieves a slightly higher preliminary RMSLE of 1.48, but appears to continue to consistently underestimate values.



Predictions made using the test data appear to perform significantly better than in the case of feature reduction, however, with a RMSLE of 1.66. Nonetheless there is significant room for improvement. I next explored a sparse PCA transformation to see if any improvement in performance was evident, but found a significant reduction in performance. My last model relied on the linear PCA output, but using an XGBoost predictor.

## Model Three: Linear PCA with XGBoost Regression

Having found a linear PCA approach to feature reduction perform the best, I next opted to explore the performance of a different type of predictive model. However, applying a hyperparameter-tuned model to the PCA-transformed data resulted in significantly worse performance than the random forest regression, with a RMSLE of 1.76. It appears XGBoost is also unable to overcome the challenge of data sparsity.

## Conclusion

Ultimately, I was not able to meet my benchmark score of 1.47. I was not able to overcome the extreme sparsity of datapoints, resulting in near identical predictions for the majority of observations that were consistently below true values. My next approach would be to impute the zero values in the dataset, but computational limitations prevent me from doing so at this time. Although I resisted using the leak in the data to build a model, that may have been necessary to build a model that accurately captured the dynamics of the data.