Fan Season Ticket Holder and Rejecter Segmentation & Classification

Udacity Machine Learning Engineer Nanodegree Capstone

Tejas Bala

Table of Contents

[Introduction 2](#_Toc50201888)

[Problem Statement 2](#_Toc50201889)

[Datasets and Inputs 2](#_Toc50201890)

[Data Wrangling 3](#_Toc50201891)

[Data Cleaning & Exploration 4](#_Toc50201892)

[Fan Segmentation: Clustering Analysis 7](#_Toc50201893)

[Season Ticket Buyer Classification: Supervised Learning 10](#_Toc50201894)

[Improvements and Future Steps 12](#_Toc50201895)

# Introduction

I currently work within the Business Intelligence Department for the New Orleans Pelicans basketball team. The department focuses on analytics that span ticket sales, digital media, marketing and corporate partnerships. We constantly strive to provide our sales team with new tools and analysis to make their efforts more efficient: one major project there is lead scoring. There are several approaches to a problem like this, one being a Recency-Frequency-Monetary Value model and another being a likeliness model that leans on machine learning.

We have a dataset that tracks Customer Journey across several main data entities: SeatGeek tracks primary and some secondary ticket sale data, CRM tracks sales representative interactions, Marketo tracks email and web interactions (when captured), Yinzcam tracks mobile app interactions and Fanatics tracks online merchandise purchase behavior.

I have access to a dataset and approval to use it for the purpose of a machine learning model for my capstone for this course. I have been requested to not use any Personally Identifiable Information and to only provide a sample of the rest of the data. I hope that this is not an issue if notebooks and documentation are available for evaluation.

# Problem Statement

Predicting revenue and product demand is a core requirement for optimizing efforts and maximizing profits. In sports, ticket packages are the equivalent to a subscription whereas your single game is a one-time purchase and it is a constant process to try to assess and convert our single game buyers into plan holders.

How can we predict what a single game buyer’s likelihood is to buy a ticket plan? Can we segment our Season Ticket Plan Holders and compare our Single Game Buyers to them to see who is the most probable to convert? Can we predict who will buy Season Ticket and reject Season Tickets?

# Datasets and Inputs

Below are previews of the Customer Journey tables from each data entity before aggregation queries.

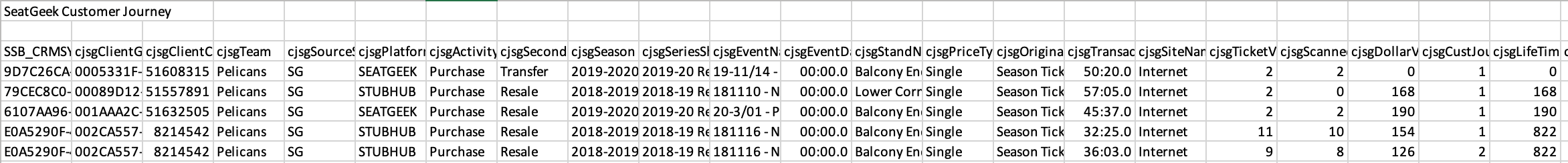


Figure - SeatGeek Customer Journey Table

A screenshot of a cell phone

Description automatically generated

Figure - SeatGeek Customer Journey Table cont.

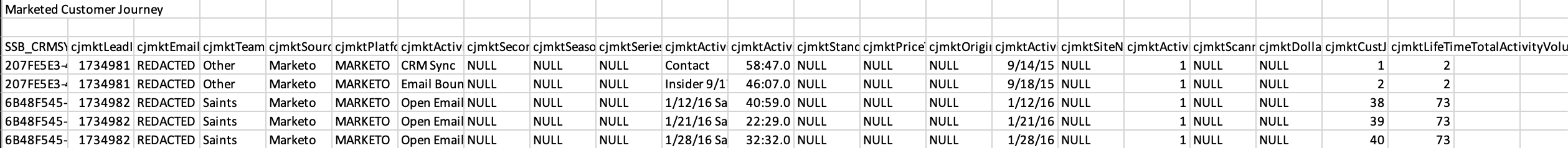


Figure - Marketo Customer Journey Table

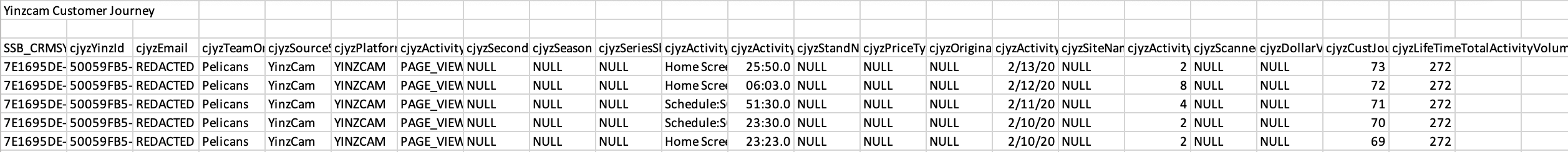


Figure - Yinzcam Customer Journey Table

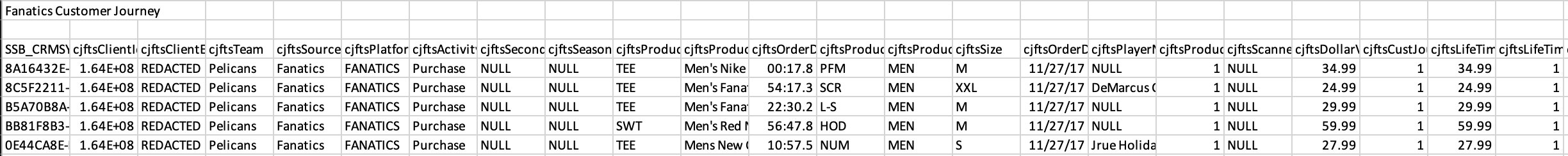


Figure - Fanatics Customer Journey Table

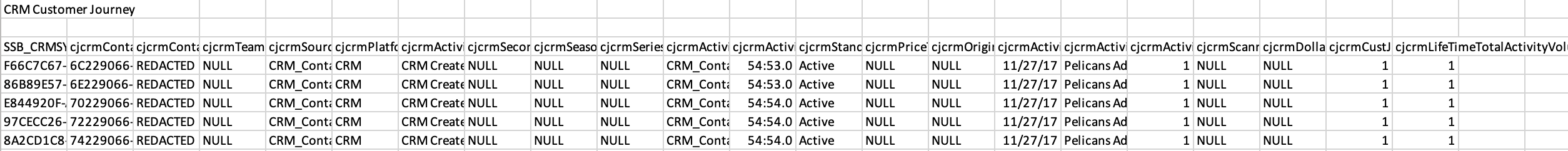


Figure - CRM Customer Journey Table

A screenshot of a cell phone

Description automatically generated

Figure - Plan Rejecter List

# Data Wrangling

To see the queries, reference localLibrary\_dataQueries.py.

The desired output was 3 subsets of data: STM (Season Ticket Member), Lost (Plan Rejecters or Lost Accounts), nonSTM (Ticket Buyers that are neither STM or Lost).

The Customer Journey tables are grouped by major activity type across data entities and metrics aggregated.

STM data set are aggregated on all activity before the date of their first Season Ticket product. nonSTM data aggregates all activity since there is no date of purchase yet. Lost data aggregates all activity as well to capture those fans that continue to buy game tickets but have rejected season ticket plans.

To see the execution of queries reference data\_acquisition.py.

The extracted CSV files were then uploaded to an S3 bucket.

# Data Cleaning & Exploration

The first major step in cleaning and transformation is merging the datasets from different entities (SG, MKT, FTS, YZ, CRM) together per fan type (STM, nonSTM, Lost). From my prior knowledge of the data, I understand that the universe of fans is much smaller in SeatGeek than in Marketo or CRM, while I am unsure about Yinzcam and Fanatics. Either way we are specific to the universe of ticket buyers meaning across the data entities we only take the fans that are present in the SeatGeek dataset.

One important step with the SeatGeek data since it is in the format below where a fan can have multiple rows based on activity type and ticket type is to pivot so that each fan has one row with columns that represent all activities and ticket types. After the pivot is complete we take the fan’s absolute minimum SeatGeek date of engagement and absolute maximum.

|  |  |  |  |
| --- | --- | --- | --- |
| SSB\_CRMSYSTEM\_CONTACT\_ID | Activity Type | Ticket Type | Metrics… |
| 12345 | Purchase | Primary |  |
| 12345 | Sell | Primary |  |

Figure - SeatGeek Before Pivot

Next step for the data is to clean up the date columns. These columns are in the format “YYYY-MM-DD HH:MM:SS” and we would like to transform them into a quantifiable value for model ingestion. A straightforward approach that dignifies the linear nature of the dates is to convert to in as is (YYYYMMDDHHMMSS). Once this is done we can create a feature for time difference that represents the length of engagement that a fan has within a data entity.

Once the data features are complete we can look at all of the features and determine if we need to drop any due to majority null. The figure below shows that we should drop all columns for Fanatics, Yinzcam and Secondary Market Ticket Transactions.

A screenshot of a cell phone

Description automatically generated

Figure - Columns Percent Null

Based on my understanding of the dataset and investigation and exploration, I understand the data contains a lot of incomplete records and thus I made the decision after dropping these columns to also drop all incomplete rows in the interest of a more robust model. This was feasible because of the size of the dataset and the amount of remaining rows.

A screenshot of a cell phone

Description automatically generated

Figure - Amount of full rows

Lastly, I took a look at the feature distributions across the fan types to see if I could pull out any high-level distinctions that would help with the clustering analysis. Nothing immediately jumped out but it was helpful to reference as I continued through the next portion.

A close up of a map

Description automatically generated

Figure - STM Feature Distribution

A screenshot of a cell phone

Description automatically generated

Figure - nonSTM Feature Distribution

A screenshot of a cell phone

Description automatically generated

Figure - Lost Feature Distribution

# Fan Segmentation: Clustering Analysis

Before Cluster Analysis I standardized all of the features to give them equal weights as well as maximizing relative difference within a feature. After standardization I could perform PCA to reduce the dimension space a bit. One important note here, I chose not to remove correlated features for PCA as correlated features will group in components and strengthen the underlying data trend. See the figure below for the explained variance of the components. I determined 5 components is a good balance of reducing dimensions and explaining over 75% of the variance in the data.

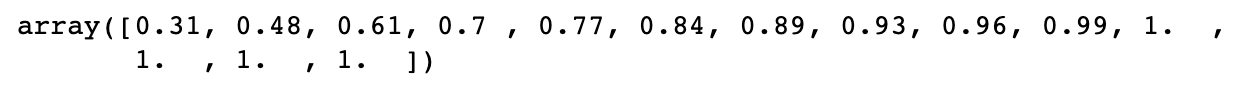


Figure - PCA Explained Variance

With PCA interpretability is very important since we are taking the raw features and turning them into components to represent the underlying data trend. See below for how this is displayed for a component.

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

Figure - PCA Explanation

For the Clustering Analysis since we have two “labelled” datasets (STM, Lost) I decided to create segmentations for both and apply our “population” (nonSTM) to those clusters.

In creating the STM Cluster, I looked at an Elbow graph to determine the proper k-value, no apparent “elbow” stuck out so I chose k=5 based on my knowledge of the domain (and also thinking about how to further explain to the sales staff 5 seemed like a good number of groups to determine defining characteristics). Before analyzing the clusters, I did the same for the Lost dataset. See below for an example of the silhouette measure and elbow graph used to determine an effect k-value.

A close up of a map

Description automatically generated

Figure - STM Cluster Elbow Graph

After applying the nonSTM data to both the STM and Lost clusters we are able to pull out some high-level insights.

As expected, the vast majority of our general population will not fall in our most popular STM buckets. One insight that immediately jumps out is that the ~100 fans in Cluster 4 we should pursue heavily since a vast majority of our STM customers are similar to them, and same for the fans in Cluster 0.

A screenshot of a cell phone

Description automatically generated

Figure - STM Cluster vs Population

Below we do the same for our Lost Clusters. This distribution looks almost identical to our General Population which makes sense but brings about the question of ‘who are our fringe cases and how can we convert them?’

A screenshot of a cell phone

Description automatically generated

Figure - Lost Cluster vs Population

For one more view at the data we take a look at the proportions rather than the raw values. In our STM graph the bars above the dashed line show the clusters that we have a better chance of converting fans into season ticket holders. Conversely, in the Lost graph the bars below the dash line show the clusters where we have a better chance of converting fans into season ticket holders.

A screenshot of a cell phone

Description automatically generated

Figure - Cluster Proportion Analysis

Lastly, I wanted to be able to understand which components and which raw features when into each cluster and how it affected the breakouts. See below for an example of a table generated for this deeper understanding.

A screenshot of a cell phone

Description automatically generated

Figure - Cluster Explanation Table

# Season Ticket Buyer Classification: Supervised Learning

For model training we took only the data from STM and Lost as we want to classify any new data as one or the other. We remain with ~6.5K records to train on, of which ~20% are STM. I kept this class imbalance in mind as I proceeded.

A picture containing table, bird

Description automatically generated

Figure - Data Size & Proportion

I held out 10% of the data as a holdout set to test my best model after a kfold evaluation of 6 models. I chose Logistic Regression, Decision Tree, Random Forrest, Gradient Boost, AdaBoost, XGBoost. This was based on my understanding of boosting algorithms and the benefits of them as well as interpretability of the model output. One concern was that my holdout, train and test sets would have different proportions of data.

A screenshot of a cell phone

Description automatically generated

Figure - Holdout, Train, Test Proportions

After running k-fold (with 5 splits) analysis, I found the average AUROC score for each model.

A screenshot of a cell phone

Description automatically generated

Figure - Model Results

The results very much shocked me, I was not expecting a model to be as accurate these are showing to be. I chose AUROC as the ROC is a common evaluation method for classification problems. I tested the model on the holdout set that also returned an AUROC of **.99**. I then tested the model on biased datasets of the holdout set filtered to just STM and just lost. Both had AUROC of ~.99.

For interpretability I also plotted the output of a trimmed Decision Tree. See below.

A screenshot of a computer

Description automatically generated

Figure - Decision Tree Output

# Improvements and Future Steps

The next step would be to predict on nonSTM and track accuracy as the sales team makes their efforts this upcoming season.

One major improvement would be to integrate some PII information: demographic and sale rep input. Additionally, having a date for when an account was “Lost” would be great and the model can turn to predicting *when* a fan will convert to a STM or Lost.

[Figure 1 - SeatGeek Customer Journey Table 2](#_Toc50201850)

[Figure 2 - SeatGeek Customer Journey Table cont. 3](#_Toc50201851)

[Figure 3 - Marketo Customer Journey Table 3](#_Toc50201852)

[Figure 4 - Yinzcam Customer Journey Table 3](#_Toc50201853)

[Figure 5 - Fanatics Customer Journey Table 3](#_Toc50201854)

[Figure 6 - CRM Customer Journey Table 3](#_Toc50201855)

[Figure 7 - Plan Rejecter List 3](#_Toc50201856)

[Figure 8 - SeatGeek Before Pivot 4](#_Toc50201857)

[Figure 9 - Columns Percent Null 5](#_Toc50201858)

[Figure 10 - Amount of full rows 5](#_Toc50201859)

[Figure 11 - STM Feature Distribution 6](#_Toc50201860)

[Figure 12 - nonSTM Feature Distribution 6](#_Toc50201861)

[Figure 13 - Lost Feature Distribution 7](#_Toc50201862)

[Figure 14 - PCA Explained Variance 7](#_Toc50201863)

[Figure 15 - PCA Explanation 8](#_Toc50201864)

[Figure 16 - STM Cluster Elbow Graph 8](#_Toc50201865)

[Figure 17 - STM Cluster vs Population 9](#_Toc50201866)

[Figure 18 - Lost Cluster vs Population 9](#_Toc50201867)

[Figure 19 - Cluster Proportion Analysis 10](#_Toc50201868)

[Figure 20 - Cluster Explanation Table 10](#_Toc50201869)

[Figure 21 - Data Size & Proportion 11](#_Toc50201870)

[Figure 22 - Holdout, Train, Test Proportions 11](#_Toc50201871)

[Figure 23 - Model Results 11](#_Toc50201872)

[Figure 24 - Decision Tree Output 12](#_Toc50201873)