Event Classification

John Kelly[1], *Cameron Snapp*[2], *David Witwer*[3]

School of Informatics, Computing and Engineering, Indiana University Bloomington

jowkelly@iu.edu, cam.m.snapp@gmail.com, gdavid.witwer@gmail.com

Abstract

This project creates a relationship between pre- and post-event data for venues with the goal of predicting event success. This research is important for helping reduce the time and effort required for small business owners to parse complicated analytics from multiple sources. Our team gathered data from Pioneer, an event venue based in Indianapolis. We preprocessed the data to double its size and reduce its features to { *weekend*, *genre*, *attending*, *interested*, *unique*, *success* }. We used several preprocessing methods: a bivariate normal distribution to create additional data points, maximum likelihood classification to identify weekend for the additional data points, and K-nearest neighbors to find *unique* for all missing values. In the end, our final classification model was a decision tree, which we will show both successfully classifies the data with a high degree of accuracy, and approximates the steps our human classifier took when making his own decisions about the data. In the future, this process can be expanded to incorporate additional pre- and post-event features to improve its utility in the industry.

1. Introduction

It is notoriously difficult for small businesses to make use of the analytics provided by digital marketing platforms. This is especially true for live event venues, which (more than other types of small businesses) are reliant on event promotion to attract customers. Although they depend upon digital marketing platforms, it is a constant challenge to use these platforms effectively, or to even understand *how* to use them effectively. Facebook, for instance, provides data on who interacts with a promoted post (i.e. a paid advertising service that spreads the word about a topic), but the actionable insights it provides can all be boiled down to ”spend more money on Facebook.”

Facebook is just one example; there are a wide variety of platforms (e.g. Instagram, EventBrite, Point of Sale, etc) that each have to be examined individually. One venue owner described to us having nearly a dozen browser tabs open to look at different reports, but gaining little insight from them because each one acted independently from the others. For him and the other event organizers we spoke with, the variety of analytics were extremely unwieldy but nonetheless a necessary evil. However, we believe we have the ability to simplify this process and help save event organizers time and mental energy. The problem this project addresses is how to tie in a wide variety of pre- and post-event factors to provide a coherent picture of the success of an event - from the marketing to the type of event to the end result - and clarify the cacophony of data coming from a variety of sources.

For some time before the beginning of this project, our team had been in communication with Pioneer, a small business in Indianapolis. Pioneer is a combined restaurant, bar, and event venue. Hosting successful events is a necessary component of their success as a business. They agreed to assist us with this project by giving us access to their data and classifying events so that we could work to identify relationships between marketing platform insights and actual event success.

Our goal was to predict event success or failure using a limited scope of data taken as a proof of concept. If we could create a predictive model for event success in a controlled environment, we could show that we can identify a relationship between disparate data elements and work to build a more complete model in the future.

2. Building the Feature Set

We began by taking event attending and interested metrics from Pioneers Facebook page. This is a practical starting data set because Facebook is the primary online marketing method for small music venues. Moreover, we had independently collected actual attendance numbers for a sample of events, allowing us to maintain consistent units rather than having to convert from dollars to attendance or vice-versa.

Initially, to identify which events had an advertising expenditure, we pulled a large number of features from Facebook. However, most of the features we could pull from the Facebook Graph API relating to marketing proved to be too sparse, so our team opted to exclude them for the purpose of this experiment. In the future, we hope to incorporate the Facebook Marketing API (a separate service) to add marketing-related features.

2.1. Transforming Original Features

In total, we ended up with 5 features from Facebook that were complete enough to use: title, date, attending, interested, and description. These features were complete for each of the 217 events we pulled from Pioneer. However, we wanted to transform these features into ones that would be more relevant for event classification.

2.1.1. Getting the Genre of an Event

From the title and description we parsed a new feature: ”genre.” The genre was the category of the performance, so it was a categorical attribute with one of the values in the set { hiphop, rock, latin, electronic, other }. This process was done manually, by researching the performing act(s). *Latin* had an extremely small sample, so we chose to merge this genre with *hiphop* to reduce the number of categories. *Hiphop* and *latin* had similar numbers for *attending* and *interested*, so it made more sense to make this merge than to merge *latin* with *other*. Also, the merge made more sense from personal experience, because these shows were more similar than latin nights are to comedy nights. In the end, we had approximately 7% hiphop, 41% electronic, 24% other, and 27% rock.

2.1.2. *Time of Week Classification*

We also parsed the *weekend / weekday* field from the date of the event. We did this using a Python script that found the day of week, and then classified an event as ”weekend” (denoted by 1) if it was on Friday or Saturday, and as ”weekday” (denoted by 0) if it was on any other day. This created a nearly even split, with 54% of events on weekends and 46% on weekdays.

2.2. Adding the Post-Event Feature

Another feature we included was the ”actual attendance” (which we called ”unique” as in ”unique guests” to differentiate from the Facebook attendance number). This feature had previously been collected manually by our team, but we only had it for approximately 40% of the data set. Nonetheless, it was a critical feature because it was our post-event data. Having post-event data is critical to our experiment because our goal was, again, to build a relationship between pre- and post-event features to identify the success of an event. However, the sparsity of this data necessitated preprocessing in order to be able to use it effectively.

2.3. Feature Cleanup

After parsing these additional fields, we removed *title*, *description*, and *date* from our feature set. This left us with the feature set { *weekend*, *genre*, *attending*, *interested*, *unique*, *success* }. *Attending* and *interested* values were completely populated. *Attending* values fell in the range (0, 234] and *interested* values were in the range (0, 444]. *Unique* was partially complete and fell in the range (0, 424]; filling this in was a step in our preprocessing. *Success* was entirely incomplete (i.e. empty). Once all other preprocessing was complete, we asked Pioneer to fill in the values for this feature however they thought best, completing our data set.

3. Preprocessing

We distinguish preprocessing from the previous section, ”transforming original features,” by whether or not we had use machine learning methods to add the data. That is, when we transform a feature, we simply take a value we know for certain and change its format (e.g. from a paragraph description to a single word genre category). On the other hand, when we preprocess the data, we take data we know for certain and estimate additional data based on that starting set (e.g. filling in missing *unique* values based on our existing ones).

3.1. Generating More Data Points

In our preprocessing, we first decided it was necessary to increase the size of the data set to make it large enough to achieve meaningful results. Our starting data set had 217 points, and we chose to double this to 434 points. Using the methods we chose, we could have made our data set arbitrarily large by generating even more points, but settled on doubling the initial set because we did not want to ”drown out” our original data. We also thought it to be important to keep the same genre ratio (and spread of feature values over each genre), which is why we chose an exact factor of two rather than some fractional increase.

Visually, the data appeared to have a log-normal distribution, but we found that using a normal distribution worked well for generating points that fit the original distribution. In line with our goal of keeping the same genre ratio, we split the data into four groups, one for each genre. In general, our method of point generation was to create a distribution for our data, and then use that distribution to create a random point within it.

3.1.1. *Using a Univariate Normal Distribution to Generate Points*

The first method of generating more points used the *interested* feature to create each genres normal distribution. We chose the *interested* feature because it had a larger range than the *attending* feature. For each of our genres, we found that genres mean and standard deviation for *interested*, then generated normal random numbers within that distribution. Any time we generated a negative value, we would discard it. We created a number of samples for each genre equivalent to however many points were categorized as that genre. After making randomly sampled *interested* values, we used linear regression to plot a relationship between *attending* and *interested*. Using the line of best fit, we then generated *attending* values to pair with each new *interested* value. We then added a random error in the range of [-RMSE, RMSE], where RMSE was the root of the mean squared error. The results of this method can be seen in Figure 1 below.

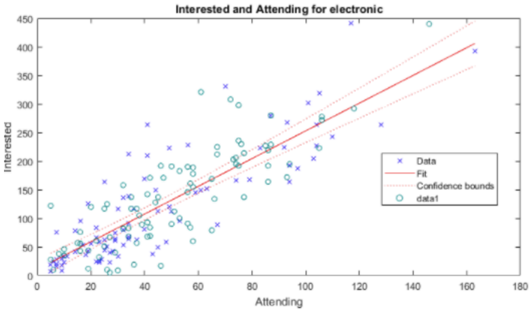

Figure 1: *interested vs. attending values for the electronic genre after point generation by univariate normal distribution.*

In figure 1, the green circles represent the generated points, whereas the blue Xs are the original points.

At first we were pleased with this method because of how closely everything fit together and how much denser it made the scatter plot. After consideration, however, we realized we were modeling the line of best fit too rigidly, creating less variance than was present in the original distribution. We believed this would have the effect of reinforcing some of the more mundane observations we could make, while simultaneously being a less accurate real-world model (which, from our data set, we observed to have a high degree of variance). Our dissatisfaction with this method led us to look for other ways to generate new *attending* and *interested* values.

3.1.2. *Using a Bivariate Normal Distribution to Generate Points*

We next tried a bivariate normal distribution to generate points, using both *attending* and *interested* feature values to create a normal distribution. Once again, we split by genre and created a separate distribution for each one. For each group of points, we found the mean vector and covariance matrix for the two features. We used those values to create a bivariate normal distribution, then once again generated a number of points equal to the size of the original genre group. If either of the generated values for *attending* or *interested* were negative, we would discard that pair of values and try again. With the bivariate distribution, we found both values, so there was no need to use linear regression for anything (unlike our trial in 3.1.1 above).

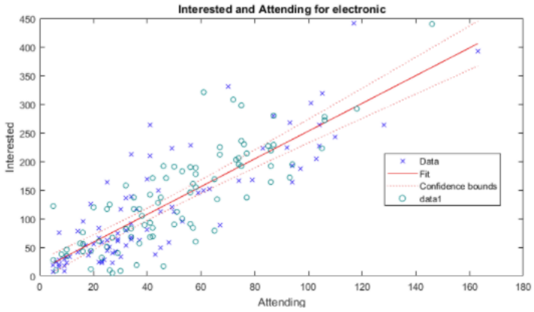

Figure 2: *interested vs. attending values for the electronic genre after point generation by bivariate normal distribution.*

As shown in figure 2 above, this created attending and interested values that deviated much more greatly from the line of best fit for the original data, but had variance that was much more closely in line with the original data set, complete with confounding outliers. Although the results were ”messier,” we thought this would be preferable and would help us find more interesting results.

3.2. Classifying New Points as Weekend or Weekday

Following point enumeration of the new data points attending and interested attributes, weekend/weekday classification was performed. We used maximum likelihood (ML) classification to obtain realistic values for the *weekend* feature for each of the generated data points. We then made two bivariate pdfs, one composed of weekend points, and the other composed of weekday points. We used the known *attending* and *interested* values as the variables in the bivariate pdf. The weekday pdf was labeled H_0 and the weekend pdf was labeled H_1.

For each generated data point in the genre, X, we calculated $P(X|H_0)$ and $P(X|H_1)$. If $P(X|H_1)$ was larger then $P(X|H_0)$ then the data point was classified as a weekend event, otherwise it was classified as a weekday event. This method was capable of classifying our training data with 70% accuracy, which was better than both ML classification without splitting by genre as well as random classification.

3.3. Using KNN to Estimate *unique* Values

With the generated data points dichotomized as either weekend or weekday events, their last missing value was actual attendance (i.e. *unique*). In order to estimate actual attendance, we first separated the weekend and weekday events into two different data sets. In both data sets, we labeled points that already had actual attendance counts as training data points. On the other hand, points that required actual attendance estimations were labeled as test data points. We estimated actual attendance through an application of the K-Nearest Neighbors (KNN) algorithm. We wanted the estimated actual attendance counts to be near that of similar data points. More specifically, if a test data point and training data point possessed similar *attending* and *interested* values, we wanted their *unique* values to be similar as well. Thus, for every test data point, i, our algorithm did the following:

- Calculate the Euclidean distance from i’s *attending* and *interested* values to that of every training data point
- Determine the 3 nearest neighbors (Note: We settled on a K-value of 3 after noticing the elbow point in figure 3.)

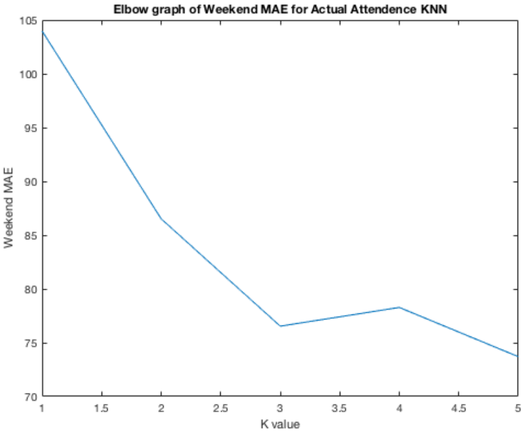

Figure 3: *Mean Absolute Error vs. K.*

- Average the 3 nearest neighbors’ *unique* values
- The average calculated in the previous step becomes i’s *unique* value estimation

At first glance, we were happy with the estimations returned by our KNN algorithm. However, we wanted to further evaluate the quality of our estimates via an error metric.

Initially, the error metric we chose to evaluate our algorithms performance was Sum of Squared Error (SSE). To determine its error, we ran our algorithm on the weekend and weekday training data sets, and computed the SSEs. For both weekend and weekday shows, the SSE was very high, which was alarming. However, after considering our data, the high SSEs made more sense. For example, assume you have a data set of 100 events. Also, for each event, assume your estimated attendance differs from the true attendance by 10 people. In turn, the SSE is 10,000, which leads one to believe the estimations are inaccurate.

Our group however, decided that an estimation that was off by ten attendees was acceptable. For example, say you are the manager at a venue, and you expect an audience of 200 people for a particular show. If only 190 people actually attend the event, you may be disappointed but you will probably not be shocked. Therefore, our group sought out an error metric better suited for event attendances, and we settled on Mean Absolute Error (MAE). We preferred MAE because it displayed, on average, how far our estimations deviated from the true attendances. For the weekend and weekday data sets, the MAEs were 76.54 and 25.37 respectively. The MAE still showcased high error, but the values made more sense for our data.

Error was exacerbated by the size of the data sets and unexpected actual attendances. The weekend and weekday training data sets contained 52 and 35 data points respectively. Therefore, there were not a lot of data points to base our KNN estimations on. Also, the training data sets possessed outliers that skewed the estimations. For instance, one weekend show had Facebook attending and interested values of 94 and 193, but its actual attendance count was 424. Another example was a weekday show that had Facebook attending and interested values of

18 and 58, but the actual attendance was 129. Considering the size of our data set and the outliers, we were satisfied with the estimations returned by our KNN algorithm. Figures 4 and 5 below display *unique* values for the weekend training and test sets.

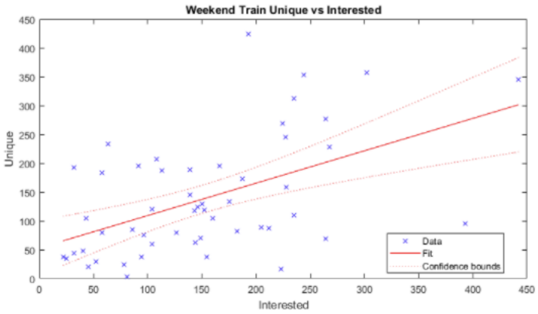

Figure 4: *unique vs. interested for the weekend training data.*

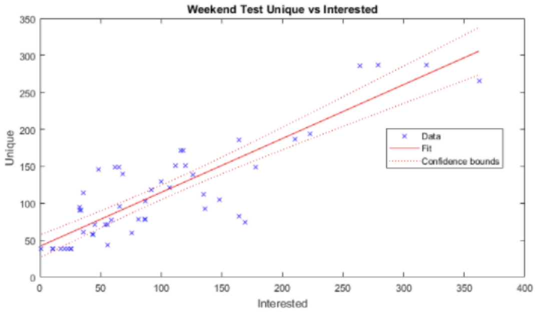

Figure 5: *unique vs. interested for the weekend test data.*

3.4. Manually Classifying *success* for all Data Points

Through analysis of the weekend or weekday classification, actual attendance, genre, and Facebook attending and interested values, our expert in the field classified each event as successful or unsuccessful. Our contacts classifications varied case by case. If the Facebook attending and interested values were between approximately 50 and 150, he assumed that the venue had spent money both on advertising the event and booking the artist. In these cases, he looked for an actual attendance (i.e. a *unique* value) of approximately twice the *attending* value.

If the Facebook values were less than 5o but greater than 10, he classified events as successful if the actual attendance (i.e. the *unique* feature) was greater than or equal to the Facebook attending value. Our contact related Facebook values between 10 and 50 to smaller, less well-known artists that did not cost money to book or advertise for. In this bracket, a 2:1 *unique* to *attending* ratio did not hold. Rather, the desired ratio was closer to a gradient from 2:1 down to 1:1. For a smaller event, if the amount of people who said they were going to the show on Facebook matched the actual attendance, our contact believed the event was successful.

The gradient guideline also applied to events with *attending* values of greater than 150. Pioneer is a smaller venue so, despite our contacts 2:1 ratio guideline, it is difficult to classify events with *unique* values of over 200 or so as unsuccessful. However, if there were that many people attending, the rule of the venue having to spend money on promotion and talent was even more true. So instead of a 2:1 ratio, the *unique* to *attending* ratio for a particular show may still be 3:2 or 4:3.

Our contact also considered the *interested* feature, but admitted that it was less important than the *unique* and *attending* features. A high *interested* value could imply that many people saw the post on Facebook, but that no one actually intended to go meaning that money was spent (or perhaps wasted) on the promotion of a not-so-popular event. Again, however, each decision was made on a case-by-case basis, as it is done in the real world. Once we had these true classifications, our next step was to determine a viable classifier.

3.5. Our Machine Learning Classifications

Using MATLABs Classification Learner app, we trained various different models to classify our data. The learner app also generated code that allowed us to replicate the classification processes. After experimenting with several classification models, we realized the best classification accuracies were returned by a decision tree and logistic regression. The logistic regression model considered two attributes, *unique* and *attending*, and classified data points with 88.5% accuracy. On the other hand, using *unique*, *attending*, and *interested* with a max split of 20 data points, the decision tree classification accuracy was 87.8%. While its classification accuracy was slightly lower, since it incorporated more attributes, we felt the decision tree was more robust than logistic regression. We believe incorporating another attribute in the classification process is more important than a 0.7% improvement in accuracy. In turn, for future work, we believe the decision tree is the best classification model for the data. Our decision tree's confusion matrix can be seen below in Figure 6.

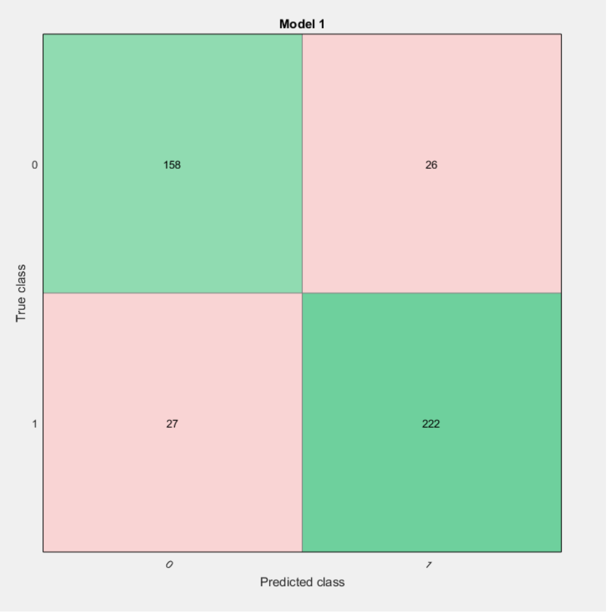

Figure 6: *decision tree confusion matrix*

3.5.1. Visual Analysis of Linear Separation

Noticing that *attending* and *unique* were the best features for predicting event success, we plotted them in a scatter plot and compared them to the line of best fit for the data. The resulting plot can be seen below in Figure 7.

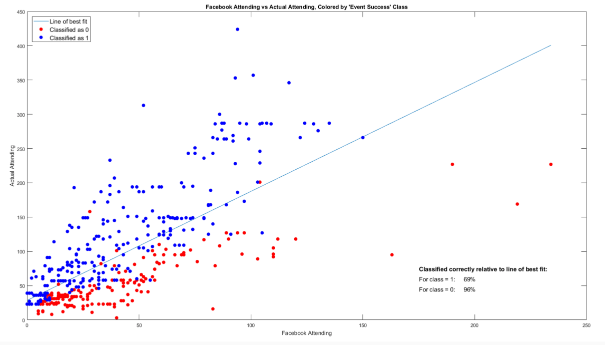

Figure 7: *attending vs. unique colored by success (blue = successful; red = unsuccessful).*

In figure Figure 7, we can see how closely the line of best fit follows the split between successes (blue) and failures (red). We can also see the 2:1 rule (as stated above) come into play, although it appears to be closer to 1.6:1 or 1.75:1 for many cases. Using just this line of best fit (above which 69% of successes lie and below which 96% of failures lie), we can achieve a classification accuracy of approximately 83%. However, our discovered method performs five points better than this naive method. We believe we can attribute this improvement to a decision trees superior ability to account for edge cases and model the human decision-making process.

4. Discussion

Over the course of this research project, we encountered both potential sources of error as well as points where we discovered our assumptions were incorrect. Our expectations were also challenged at several points during the process. Not only did we change from univariate to multivariate point generation methods and experiment with different K-values for KNN, but we also learned that not all of our features were as useful as we initially expected.

4.1. Potential Errors

Among the main potential sources of error were our small sample size, the sparsity of our *unique* feature, and the subjectivity of Pioneers own classifications.

4.1.1. Sample Size Error

To our credit, despite many venues having distinct characters, we have identified trends that we expect should scale to larger sizes. We expect this because, whether one manages a small ”indie” venue or the hottest nightclub in town, people in an area typically go out in the same range of times. Weekends will be the busiest and most profitable days of the week, and attendance will rise or fall based on the amount of people that are aware of an event. Because of this fact, we believe that we will be able to abstract our research and apply it to other venues in the future.

That said, to do so we will certainly need a larger sample moving forward to confirm this intuition and help create expected values for various statistics. In other words, when evaluating the success or failure of an event, the results will be relative to the expected outcome of the event. The expected outcome of a nightclub will be different than the expected outcome of a small venue. Because we were working with just one venue, we limited the scope of our problem to creating a model for that specific one. Although, we argue that this model could be applied to venues of a similar size or type. In addition, we also created a method for generating a model that can be applied to other types of venues, because (as stated above), the general pattern of inputs will hold - only the scale and expectations will change.

4.1.2. Sparsity of unique Potential Error

Using our KNN algorithm, we were able to fill in the *unique* column for our data set. However, given that only 40% of our initial data set had a *unique* value to begin with, we were generating *unique* values for close to 80% of our new, augmented data set. While this may seem to be an exceptionally large amount that could skew our results, we were not concerned. In our view, our task was to create a realistic data set composed of points that Pioneer could judge to be either successful events or failures. Even if we modeled the original data set with a high degree of error, we could justify our decisions at every step of the process and were confident - through visual and scientific analysis - that our estimations were within the realm of possibility.

4.1.3. Justification for Method of Point Generation

Our method of point generation was to create a normal distribution for each genre, then randomly create new points that fell within that distribution. However, a critical observer might find several reasons to disagree with this chosen strategy.

For one, visual analysis does not necessarily suggest the data is normally distributed. In the figure below, it looks as though the data might more closely follow a log-normal or exponential distribution. But an exponential distribution, of course, is a measure of intervals, and does not really make sense with our data. A log-normal distribution, however, may have been a good distribution to try. Nonetheless, we were satisfied with our results, shown in Figure 8, and did not feel the need to experiment further in this area.

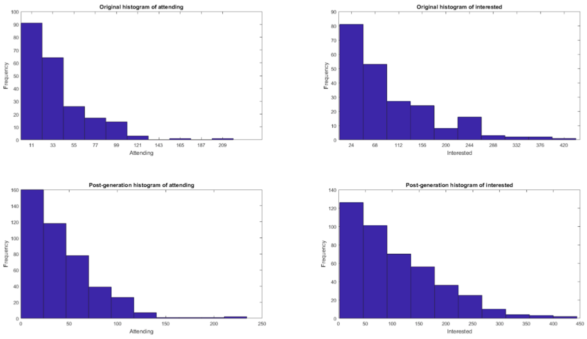

Figure 8: *pre- and post-generation attending and interested histograms.*

Another point of criticism might be that by splitting apart the points by genre, we created samples that were too small to create a reasonable distribution. While this is a valid point, it had the added benefit of adding a high-degree of variance to certain genres that were themselves highly variant (e.g. hiphop, shown below in Figure 9)

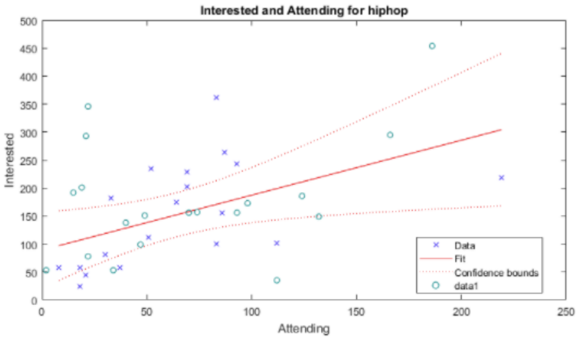

Figure 9: *interested vs. attending for the hiphop genre*

Overall we were satisfied with our methods because our primary goal, after all, was to generate a range of possible values that Pioneer would be able to look at and classify as either a success or failure. In truth, any values for *attending* and *interested* are possible - our goal was to create ones realistic enough to not appear to be false and mislead Pioneer during their classification.

4.1.4. Classification Error

When asked to evaluate his own accuracy, our contact at Pioneer acknowledged that - especially for the large events - there was the chance of additional context (in terms of sunk costs or other incentives) that would change his classification. Based on his personal experience and expertise, our contact estimated that this happened less than 10% of the time. Given this estimate, we judge his own classifications to have an error rate of about 5%.

4.2. Changes in Expectation

We originally expected the *weekend* and *genre* features to be relevant for our decision tree classifier. We were surprised to learn that not only was this not the case, but also that the *interested* feature was only used a small amount. The *interested* feature was used as sort of a tiebreaker in certain instances. However, that is not to say that it was a waste of time to look at these features. Specifically, the genre helped us augment our data set in a way that mirrored the original set, and we split our data on the *weekend* feature when we used KNN to estimate *unique* values. Despite their omission from both our decision tree and our contacts own description of his methods, we anticipate these features being useful in the future for predicting *unique* values in practice.

4.3. Future Work

We are eager to continue this work with Pioneer and potentially other venues. Armed with a larger data set, we hope to refine our methods and improve our classification accuracy. Along with more data points, we also hope to incorporate more attributes. For instance, by incorporating event financials, we aim to fix the potential classification error our contact at Pioneer was concerned about. Knowing the costs and income of an event will undoubtedly help in classifying large events as successful or unsuccessful. Through use of Facebook's Marketing API, we would also like to further analyze the Facebook posts used to advertise events. The number of clicks, comments, and shares a post receives could also correlate with event success.

Along with their correlation to event success, we are curious to see how Facebook posts advertising events gather interest. By parsing, tokenizing, and stemming the language of a post, we would like to analyze whether certain words foster interest. The results from this experiment could help venues develop ad campaigns that lead to higher event attendances.

We are also interested to see if the accuracy of our KNN algorithm improves with a larger data set. If our KNN algorithm provides better attendance estimates, it could be quite useful for venue managers. With an accurate attendance predictor, managers could make wise staffing and inventory decisions.

Finally, we know that venues use other tools in addition to Facebook. In tandem with bringing in financial records and advertising expenditure, we are intrigued by the possibility of including data from APIs such as Instagram, EventBrite, or Square, to further complete our feature set.

5. Acknowledgements

In this research project, we do not include references to past work, but we would like to take this opportunity to acknowledge the outside assistance we that helped us complete our work. We would like to thank Donald S. Williamson for providing lectures slides that aided our research. We would like to acknowledge the Classification Learner App, which was instrumental in the implementation of our decision tree and logistic regression experiment. We would like to acknowledge that we obtained data from the Facebook API. In addition, we accessed information from Wikipedia and the MATLAB documentation. Finally, this project would not have been possible if it weren't for Pioneer's open-mindedness. Many thanks to them, as well as for all the time and effort our contact, Wesley Cook, devoted to help us get an A.