Will Your Event Sell Out? Answering a Prediction Problem Using Machine Learning Algorithms

ECON 483: Economic Applications of Machine Learning

Daniel Bathaei Amanda Muirhead

August 15, 2023

While by itself it is a piece of worthless paper, a ticket grants individuals access to an event that is otherwise closed to the public. If an event is well-liked, many people will buy tickets for it. If a performer, sports team, or exhibition achieves fame and popularity, more people might want to attend their events than are slots available. Once every ticket for an event has been purchased, that event is sold out. To performers, selling out is a sign of popularity. To producers, it is a sign of a missed opportunity. They could have charged more for their tickets, booked a bigger venue, or offered VIP packages. The ability to accurately anticipate whether an event will hit full capacity can have significant implications on a business' demand forecasting, market analysis, and revenue optimization, not to mention the subsequent effects on consumer spending and labour demand. Consequently, producers are inclined to wonder whether a set of variables such as price, date, and location can predict the likelihood of their event selling out. To this end, we use machine learning models to predict the results of such a question.

The onset of machine learning in economics, with its capacity to discern patterns from intricate data, has become an indispensable tool in tackling complex prediction problems. Though it has several varying definitions, the study of machine learning boils down to computer programs' abilities to learn and adapt to new data without the interference of humans (Babenko et al. (2021)). To predict the likelihood of an event selling out of tickets, we will employ several machine learning techniques, namely Classification Trees, Logistic Regression, and Random Forests.

Background and Current Literature

Event tickets have been in the news a lot lately, namely for the escalating rates at which they are being priced. Event-goers around the world are increasingly lamenting over the rising costs of tickets, as reports come out of potential price gouging and fees as high as 78% of the original listed price (*Why Everyone's Mad at Ticketmaster Right Now* (2022)). While in 2018 it was reported that ticket sales had doubled since the 1990's ("Gig Ticket Prices Have Doubled Since 1990s" (2018)), the percentage increase is now estimated to be near triple their pre-Y2K levels (LastWeekTonight (2022)), proving that forecasting event demand is already one of many lucrative strategies being employed by artists and ticket-distributing giants such as Ticketmaster. Currently, event tickets sales are a $78bn USD industry, expected to show a growth of 9.7% by 2024 (*Event Tickets - Worldwide | Statista Market Forecast* (n.d.)).

To contextualize our research, we studied existing literature surrounding event sell-out and other related quantity demanded predictions. Previous studies have illuminated various factors that contribute to event success, encompassing variables such as ticket pricing, marketing strategies, historical attendance data, and socioeconomic indicators (Krueger (2005)).

Limitations

While writing this paper, we observed that companies work very hard to conceal any information related to sales volume and quantity of tickets demanded. StubHub has stopped providing businesses and researchers with access to its application programming interface (API), only provides API services for producers for their own posted tickets[1] Ticketmaster has recently removed any indicator of sales volume on their API database. Companies like Pollstar[2] charge considerable fees before providing any information on ticket sales. This further solidifies the current importance and relevance of ticket sale information.

Methodology

We use publicly available resources[3] to prepare the data for our models. Pursuant to our objective, we collected data from SeatGeek using the website's API. Similar to other websites, SeatGeek has reduced their available sales volume data; however, SeatGeek still provides some insight on sales volume. Although the company has announced their plan to move away from disclosing sales volume information through their API altogether, this plan is yet to be implemented.

SeatGeek

SeatGeek currently provides information regarding the volume of tickets sold by an event, performer, and/or venue. This information is shared through the 'score' variable, a scaled[4] measure of an event, performer, or venue's ticket sale volume relative to their type (i.e., music festival, comedy show, etc.). This means that if an event sells more tickets than any other event similar to its type, that event will receive a score of 1. If an event makes no sales, that event will have a score of 0. For each observation we can gather an event score, venue score, and an individual score for each performer within that event. This information is highly valuable as it is derived from the quantity of units sold. Conveniently, SeatGeek analyzes the social media following of each event's performers, and provides a 'popularity' value based on how popular an events performers are, also scaled between 0 and 1. Consequently, the variables we gather from SeatGeek are shown on Table 1.

Table 1: Identifying the Class and Type for each variable in the final dataset.

Variables	Class	Type	Variables	Class	Type
id	integer	integer	listing_count	numeric	double
date	character	character	average_price	numeric	double
local_date	character	character	lowest_price	numeric	double
score	numeric	double	highest_price	numeric	double
announce_date	character	character	median_price	numeric	double
type	character	character	segment	character	character
n_performers	integer	integer	primary_performer	character	character
popularity	numeric	double	prim_perf_event_count	integer	integer
venue	character	character	prim_perf_score	numeric	double
country	character	character	performer_2	character	character
state	character	character	performer_2_event_count	integer	integer
venue_score	numeric	double	performer_2_score	numeric	double
v_n_upc_events	integer	integer	sold_out	integer	integer

[1]https://developer.stubhub.com/api-reference/sales#tag/Sales

[2]https://www.pollstar.com/subscribe)

[3]Ethical obligations dictate that we only use publicly available data, even if we use scraping methods.

[4]$0 \leq score \leq 1$

Data Collection: Setup

If an event sells out of tickets, SeatGeek immediately removes that event's ticket pricing information. This posed a challenge to our objective since we require both ticket pricing information as well as the 'sold_out' status of the event. To overcome this issue, we set up our data collection strategy accordingly. We gathered information on all events for a given week on the Monday of that week. Then, at 9PM of each day, we collected data on all remaining events to collect an event's ticket pricing data and determine whether each one had been sold out. Given that our data is gathered over one week in August, our set up implies the presence of heteroskedasticity in the sample which is exacerbated if we choose to remove "event date" as a regressor.The final data set contains 2056 observations with 26 variables (see Table 1). Of those variables, 5 are the scaled scores calculated by SeatGeek.

The final data set contains 2056 observations with 26 variables (see Table 1). Of those variables, 5 are the scaled scores calculated by SeatGeek. The following table demonstrates

Table 2: Summary statistic for SeatGeek generated scores.

score	popularity	venue_score	prim_perf_score	performer_2_score
Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. :0.0000
1st Qu.:0.3180	1st Qu.:0.6380	1st Qu.:0.4800	1st Qu.:0.3300	1st Qu.:0.0000
Median :0.3810	Median :0.6700	Median :0.5800	Median :0.3900	Median :0.0000
Mean :0.4062	Mean :0.6741	Mean :0.5792	Mean :0.4199	Mean :0.1454
3rd Qu.:0.4662	3rd Qu.:0.7130	3rd Qu.:0.6900	3rd Qu.:0.5100	3rd Qu.:0.3300
Max. :0.9600	Max. :0.9600	Max. :0.9700	Max. :0.8600	Max. :0.8000

Data Collection: Preparation[5]

In its initial form, our data set possessed irrelevant, unique, and over-categorized columns, as well as incomplete rows. We first import and prepare the data using the following R code:

```r
data <- read.csv("../Week1/SoldOutData.csv")
data <- subset(data,
               select = c(
  -id,-venue, -type,
  -primary_performer,
  -performer_2, -date)) # remove irrelevant and row-unique columns.
data <- na.omit(data) # omit NA rows.

# lists to declare each column as its specific type of data.
date_columns <- c("local_date", "announce_date")
numeric_columns <- c("score", "popularity", "venue_score",
                     "listing_count", "average_price", "lowest_price",
                     "highest_price", "median_price",
                     "primary_performer_score", "performer_2_score")
integer_columns <- c("n_performers", "venue_n_upcoming_events",
                     "primary_performer_event_count", "performer_2_event_count")
factor_columns <- c("country", "region", "segment")
```

The "state" column has 57 categories composed of U.S. states and Canadian provinces. As several R functions do not have the ability to handle a 57 category column, we use the U.S. Census Bureau's definition

[5]We used python throughout the process of sending API requests and managing the data. To replicate the process, please visit https://github.com/dbathaei/econ483.

of Regions to combine geographically similar states and reduce the number of categories to 4 regions: "West", "Midwest", "Northeast", and "South" (*Census Regions and Divisions of the United States* (n.d.)). We also placed Canadian provinces in these 4 categories according to their location. We then erase the irrelevant portions of the data set and convert each column to its appropriate type.

Prediction Models

This paper attempts to solve a classification problem where a class label is predicted given our input data. The dependent variable is binary and can only hold the values 1 and 0, therefore, we are predicting the likelihood of the dependent variable being equal to 1, or sold out. We employ models that we will answer the following general equation:

$$\beta_{\text{sold\_out}} \times I_{\text{sold\_out}} = f\left(\sum_{i}^{d} \beta_i \cdot X_i\right) \tag{1}$$

where $\beta_{\text{sold\_out}}$ represents the likelihood of the `sold_out` variable being equal to 1, i represents a set of covariates in the `data`, and f represents some form of transformation–if any–performed on the covariates.

In our prediction models, we aim to maximize the variance in the sample's response variable that the model accounts for in its predictions while minimizing the likelihood of overfitting to the sample. Overfitting is defined as the production of a model that corresponds too closely to a specific sample, thus losing its accuracy in predicting future observations outside of our sample data. We combat overfitting using various techniques based on each respective model we use.

Linear Model

First, we examine the residuals of the linear model to better understand the nature of our sample:

We observe a clear downward sloping pattern in the Residuals vs. Fitted Values plot, showing that the residuals decrease as the fitted values increase. This identifies that the model is inaccurate in capturing the relationship between the variables, making a linear model inconsistent. However, we may still examine whether selecting a smaller subset of covariates will help reduce overfitting for our non-linear models.

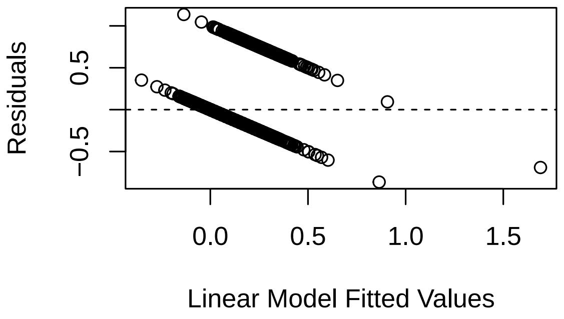

Figure 1: Residuals Plot for the Linear Model

Best Subset Selection

The residual sum of squares (RSS) is defined as the sum of the discrepancies between the actual data and the predictions of a model. In other words, the RSS measures how well the model fits the data. Best subset selection aims to achieve the lowest RSS with the fewest number of variables by creating a penalty term for each extra regressor used in the model. In this paper we separately use the Bayesian Information Criterion (BIC), Adjusted R^2, and $C_{(p)}$ as the information criteria for our subset selection:

```r
BESTSUBSET <- regsubsets(sold_out ~., data = train_data, nvmax = length(colnames(train_data)))
BSM <- summary(BESTSUBSET)
data.frame(
  Adj.R2 = which.max(BSM$adjr2),
  CP = which.min(BSM$cp),
  BIC = which.min(BSM$bic)
)
```

```
##   Adj.R2 CP BIC
## 1     12  9   5
```

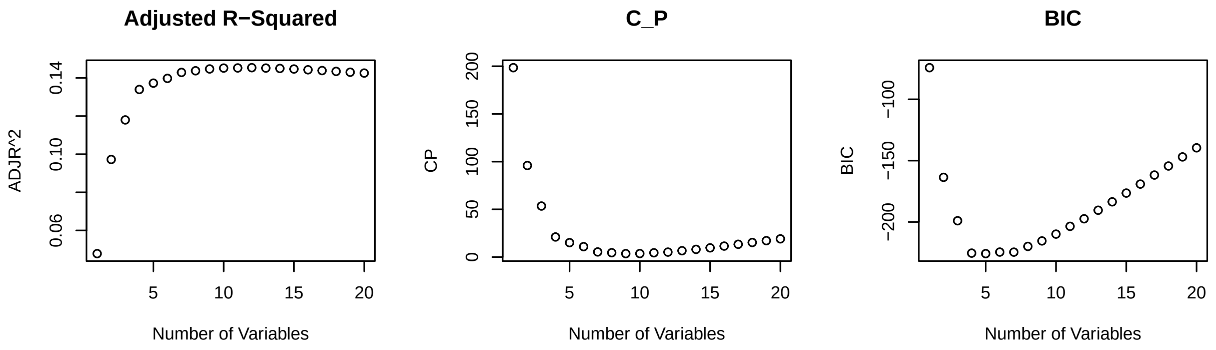

Figure 2: Best achieved information criteria scores for the optimum combination covariates given different numbers of allowed regressors.

The following R code provided us with the best variables for our prediction models based on each information criteria. Given that ours is a classification problem, we use the logistic model to cross-validate the accuracy of our best subsection selection. We regressed the `sold_out` column 4 separate times: once on all the available covariates, and once for each set of selected covariates displayed on Table 3.

Table 3: Best selected subsets based on each criterion

Adj.R2	Cp	BIC
local_date	announce_date	announce_date
score	n_performers	venue_score
announce_date	venue_score	lowest_price
n_performers	lowest_price	median_price
popularity	median_price	segmenttheater
venue_score	segmenttheater	
lowest_price	primary_performer_event_count	
median_price	primary_performer_score	
segmenttheater	performer_2_score	
primary_performer_event_count		
primary_performer_score		
performer_2_score		

Based on Table 3, we can observe that SeatGeek's `score` covariates hold strong predictive power.

Logistic Model

Given that our response variable can only be 0 or 1, we have to assume that probabilities fall within this range. A linear model may improperly predict that an event will sell out with a negative probability, or a probability of higher than 100%. One method to to remedy this is the logistic model that ensures that the predicted probabilities fall within 0 and 1. The logistic model leverages the logistic function to constrain its predicted probabilities within the bounded interval, and employing the logistic model ensures that the continuous probability estimates are confined to this specific range, thus avoiding possible issues that might arise from linear models and providing a more suitable framework for binary response variables.

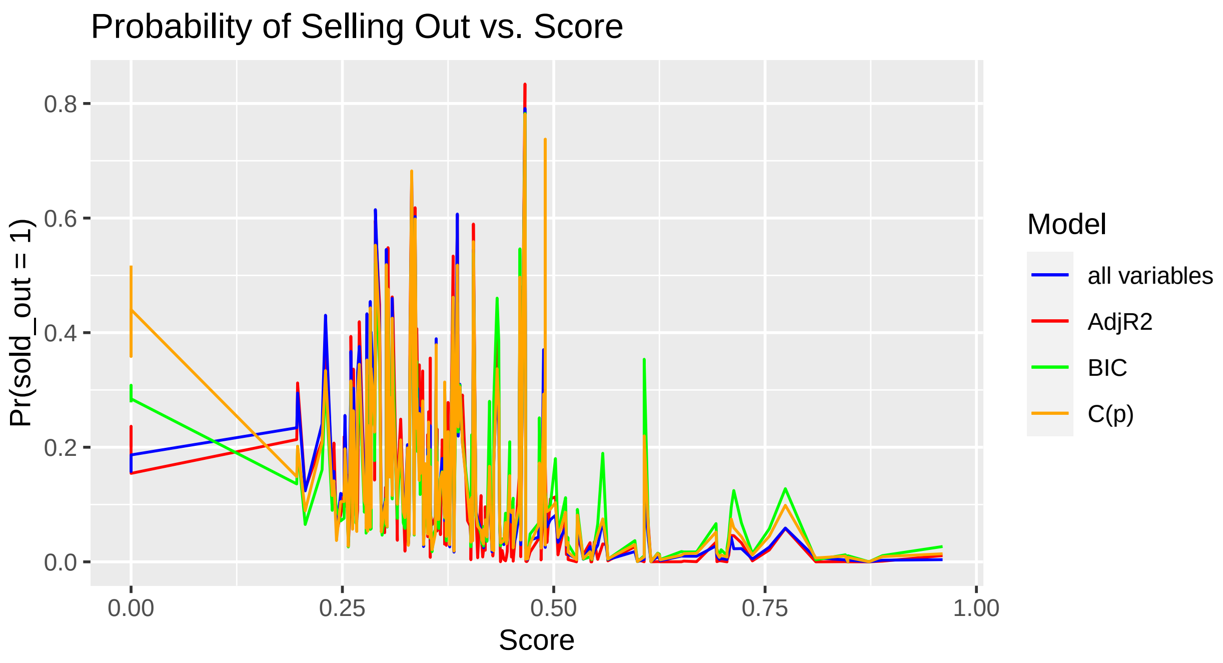

Figure 3: Graph of Logistic Predicted sold_out Probabilities for Different Subsets.

Upon reviewing the Figure above, we observe that the likelihood of selling out not only does not increase as the `score` and `venue_score` variables for that event increase, but that highly popular shows are suprisingly less likely to sell out.

However, it is important to note that the Logistic comes with several assumptions. The independence of errors, absence of multicollinearity, as well as strongly influential outliers may make this model unsuitable for our data, given that all `score` variables are derived from overlapping ticket sales information.

Classification Trees

Classification or Decision Trees, on the other hand, do not rely on such parametric assumptions, can capture more complex patterns in the data, and can address strong outliers through 'pruning'. We began with a complex and unpruned tree, and then used the complexity parameter table to derive the best framework for pruning. We then tuned the model according to the best complexity parameter. The below R chunk represents these steps[6]:

[6]In the Rmd file, there are available examples to produce a graph for the unpruned decision tree. However, given that our objective is to find the best-pruned, cross-validated tree, that is the only one we displayed in this paper.

```r
full_tree <- rpart(sold_out ~ ., data = train_data, method = "class")
cp_table <- data.frame(full_tree$cptable)
best_cp <- cp_table[which.min(cp_table$xerror), "CP"]
pruned_tree <- prune(full_tree, cp = best_cp)
rpart.plot(pruned_tree, main = "Decision Tree for 'Sold Out' Predictions")
```

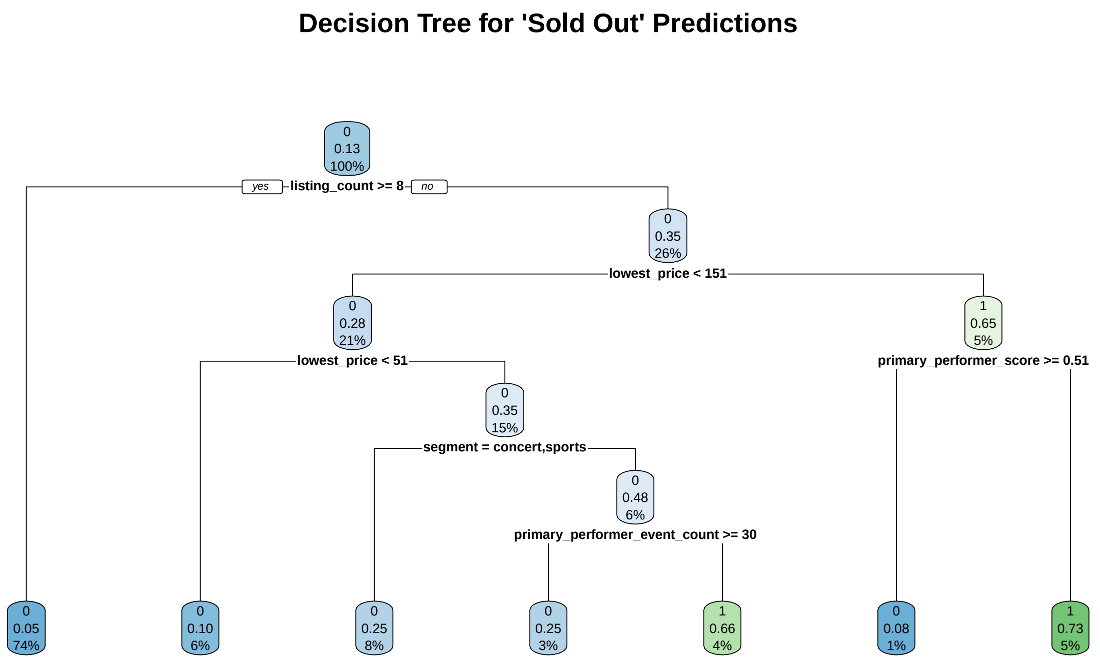

Figure 4: Best-Pruned Decision Tree Model

The figure above is derived by minimizing the complexity parameter, as well as the cross-validation error `xerror` in our model. The following graph displays the tree size with the lowest relative error:

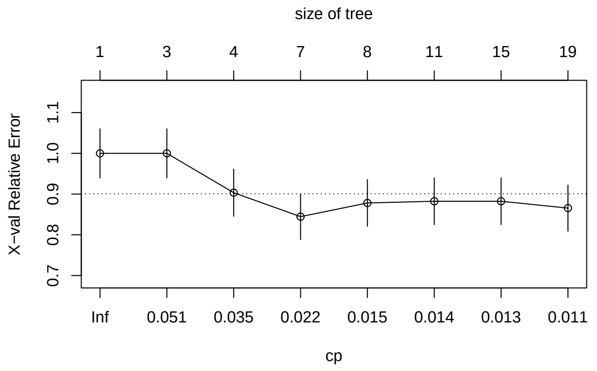

Figure 5: Graph of Cross-Validated Relative Errors

While one tree may provide insightful information, we may still improve accuracy by using multiple trees and using the majority prediction among those trees.

Random Forest

A Random Forest model begins by creating multiple subsets of the data, including some observations many times and allowing the possibility of not using an observation at all. This is known as bootstrap sampling. The model also randomly chooses a set of covariates for each subset, introducing further randomness into the model. This is known as feature sampling. The model then trains a tree based on each subset, creating a forest of Classification Trees.

Random Forest models can be tuned to follow specific criteria for how trees split the data, and what the minimum number of observations must be at each terminal node or 'leaf' in the tree. In this paper, we utilize all available splitting rules, as well as a range of minimum terminal node sizes in order to find out the best Random Forest model. The following R code represents all tuning choices we utilize for our model:

```r
num_predictors <- ncol(train_data) - 1
mtry_values <- c(2, floor(sqrt(num_predictors)), floor(num_predictors / 2), num_predictors)
RDF_tune_grid <- expand.grid(mtry = mtry_values,
                             splitrule = c("gini", "hellinger", "extratrees"),
                             min.node.size = c(1,5, 10))
```

The `RDF_tune_grid` encapsulates three sets of tuning rules that govern the construction and functioning of the Random Forest model. These tuning rules are vital in achieving the optimal balance between model complexity and predictive accuracy. Here, we'll break down the three tuning rules:

- `mtry` (feature sampling): Defines the number of predictors randomly sampled at each split. A model's robustness is enhanced by using different values for `mtry`, which ensures that the trees in the forest are decorrelated.
- `splitrule`: This rule determines the criterion used to split the nodes. Each of the three rules we have chosen measures the quality of a split in a unique way, consequently influencing the decision-making process.
- `min.node.size`: As previously mentioned, this rule controls the minimum number of observations that at each terminal node of the trees in the forest.

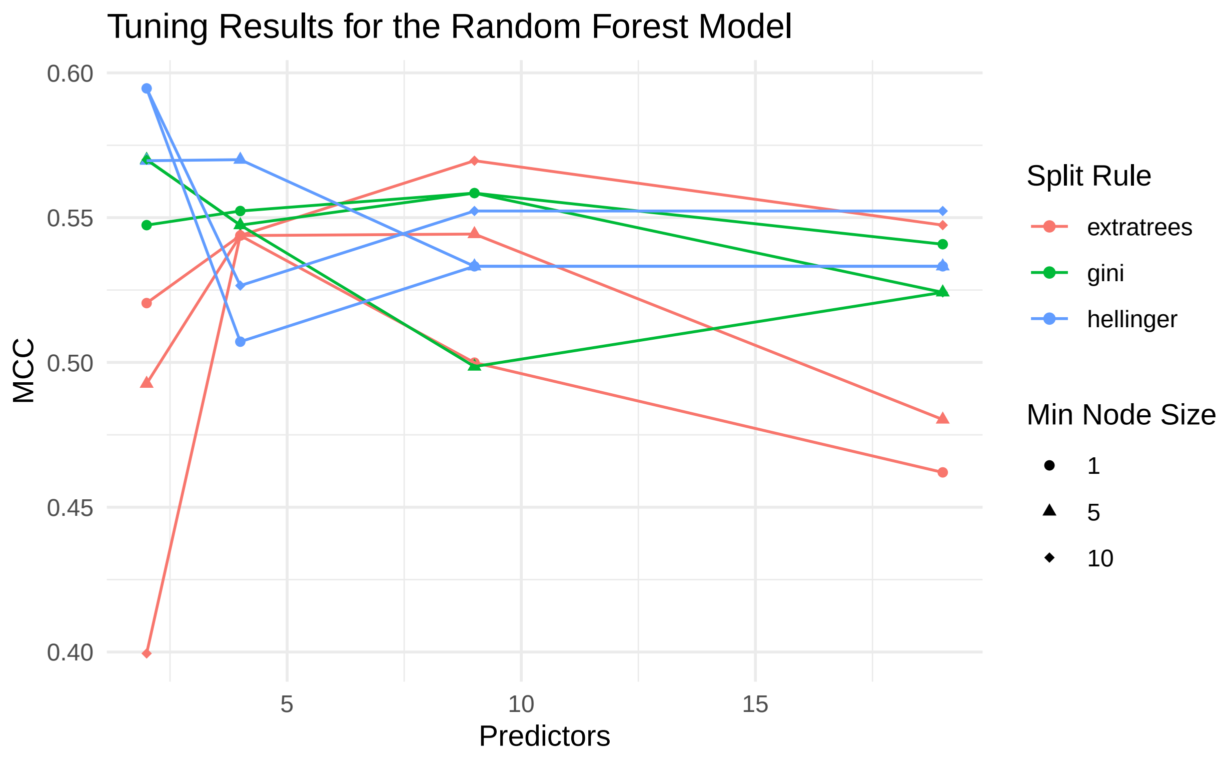

Figure 6: Graph of MCCs for different number of randomly selected features considered for split decisions (mtry).

The above graph shows that the highest Matthews Correlation Coefficient (MCC) is achieved by the Random Forest model that following set of rules:

```
##   mtry splitrule min.node.size       MCC
## 1    2 hellinger             1 0.5946334
```

Results

We generated several other models for our prediction problem. However, due to their low predictive power and lack of efficient setup, we opted to focus on the models that had better predictive power. After training and cross-validating our data on 1800 observations, we used the remaining 227 observations as our test data. The following table summarizes the resulting MCC for all models created for our prediction problem:

Table 4: All models and their corresponding MCC

Model	MCC
Random Forest - Best Model	0.5946334
Pruned Decision Tree	0.4891001
Unpruned Decision Tree	0.4721993
Logit-Cp	0.3004985
Support Vector Machine	0.2805954
Logit-All Vars	0.2768803
Logit-Adjr2	0.2600718
Logit-BIC	0.2156108

Conclusion

In our study we analyzed the dynamic nature of event popularity, which highlighted the importance of demand forecasting along with industrial organization and market analysis. Our data sourced from SeatGeek, though boasting intriguing variables such as artist and venue popularity score, may not provide a full representation of the greater population due to potential selection and temporal biases. Another limitation noted early on was the restricted nature of some data of interest. The absence of certain predictors and the subsequent quality issues underscored the need for careful consideration when interpreting our findings.

Additionally, it is important to recognize that the effectiveness of predictive models is intertwined with the parameters chosen during their formulation and training in cross validation. Parametric estimators take into account local data to the observation of interest, which grants us the freedom from restrictive parametric forms. Whereas OLS is a popular example of a global parametric estimator, the functional form of regression functions such as these runs the risk of giving us deficient predictions.

As is the case with many supervised learning methods, we were subject to some select limitations when applying parameters to our entertainment industry data. Due to these limitations we observed a noticeably lower predictive ability from the Logistic regression methods, the reason plausibly being the requirement of assumptions such as the absence of multicollinearity, strong outliers, and variance of errors.. The balance between bias and variance, over- and under-fitting, as well as the trade-offs we face in model complexity necessitate careful consideration to ensure the reliability of our findings. Thus, the method proving to be our best predictor was the random forest, borne from several Decision trees.

In conclusion, our study not only highlighted the multifaceted landscape of event popularity prediction but also underscored the intricate interaction between parameterization and predictive power. As we look ahead, refining and expanding the boundaries of predictive modeling, we look forward to adding to the insightful predictions in the dynamic domain of event management.

References

Babenko, V., Panchyshyn, A., Zomchak, L., Nehrey, M., Artym-Drohomyretska, Z., & Lahotskyi, T. (2021). Classical Machine Learning Methods in Economics Research: Macro and Micro Level Examples. *WSEAS TRANSACTIONS ON BUSINESS AND ECONOMICS*, *18*, 209–217. https://doi.org/10.37394/23207.2021.18.22

Census regions and divisions of the united states. (n.d.). https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf

Event Tickets - Worldwide | Statista Market Forecast. (n.d.). https://www.statista.com/outlook/dmo/eservices/event-tickets/worldwide

Gig ticket prices have doubled since 1990s. (2018). *BBC News*. https://www.bbc.com/news/business-42982769

Krueger, Alan B. (2005). The Economics of Real Superstars: The Market for Rock Concerts in the Material World. *Journal of Labor Economics*, *23*(1), 1–30. https://doi.org/10.1086/425431

LastWeekTonight. (2022). *Tickets: Last week tonight with john oliver (HBO)*. https://www.youtube.com/watch?v=-_Y7uqqEFnY

Why Everyone's Mad at Ticketmaster Right Now. (2022). https://time.com/6207167/ticketmaster-ticket-prices-expensive-backlash/