# Final Report
# GWCCA Redesigned Event Valuation Process

**Senior Design Team 5**
Daniel Alayo-Matos, Hailun Chang, Brandon Kang, Yunsang Kim,
Emily Kornegay, Peyton Skinner, Mayke Vercruyssen, Yihua Xu

**Team Liaison**
Hailun Chang
hchang300@gatech.edu
404-933-7518

**Client Contact**
Mark Koeninger
MKoeninger@GWCC.com

**Faculty Advisor**
Dr. Alexander Shapiro
ashapiro@isye.gatech.edu

December 2nd, 2019

# Executive Summary

The Georgia World Congress Center (GWCC), managed by the Georgia World Congress Center Authority (GWCCA), is the third largest convention center in the nation. Generating revenue by hosting events and charging for rent, food and beverage, labor services, and other amenities, the GWCCA holds around 300 events annually. With a firmwide strategy to grow profits, the GWCCA identified improving event selection as a critical target. Thus, the GWCCA is shifting their business model to assess profitability event by event instead of date by date.

Currently, the GWCCA does not have room assignment guidelines for incoming events, resulting in high operating costs due to many small events utilizing all three buildings despite only needing one building of space altogether. Additionally, their current budgeted cost is on average 61% off of the true cost, which makes it challenging for the GWCCA to evaluate potential events; 6.4% events were held at loss due to unexpected high cost. As current pricing methods do not consider potential cost and the GWCCA cannot charge client post event, the GWCCA experiences fluctuations in profit earned per event.

To help the GWCCA tackle current challenges, we designed three methodologies to be used for each incoming event; (1) Room Assignment Optimization Model that outputs cost efficient rooms, (2) Cost Prediction that returns a predicted cost, and (3) Profit Margins Classification that recommends a baseline price. To make sure the client can interact with the designs when evaluating an incoming event, all three designs are packaged in an easy-to-use web application.

With the web app, the GWCCA can enter event information that will serve as inputs to the models and interact with model outputs. Once the information is entered, the first model will run and its output will be used as inputs for the second model. The result of the second model will feed into the third method to output a recommended baseline price on the web app for the GWCCA to consider moving towards negotiation phase.

As the GWCCA will continue improving data collection in the future, our work product is expected to decrease the number of spread-out events by 19%, improve the cost prediction accuracy by 36%, and increase annual profit by $1.1 million.

# Table of Contents

**Current Business** ..... **1**
**Client Overview** ..... **1**
**System Description** ..... **2**
**Approach** ..... **2**
**Goal and Methodology** ..... **2**
**A. Room Assignment Optimization Model** ..... **2**
**B. Cost Prediction Model** ..... **4**
**C. Profit Margin Classification** ..... **7**
**Implementation** ..... **8**
**Deliverable and Recommendations** ..... **8**
**Risk and Mitigations** ..... **9**
**Value & Validation** ..... **10**
**Appendix** ..... **11**
**APPENDIX A: Data Collection and Cleaning** ..... **11**
**APPENDIX B: Room Assignment Model** ..... **15**
**APPENDIX C: Cost Prediction Model** ..... **22**
Section I Feature Introduction and Feature Elimination ..... 22
Section II Cost Prediction Model ..... 28
**APPENDIX D: Classification** ..... **36**
**APPENDIX E: Deliverable** ..... **37**
**APPENDIX F: Value Calculation** ..... **38**
**APPENDIX G: User Manual** ..... **39**
**Initial Setup Requirements** ..... **41**
**Web App User Manual** ..... **45**
**Back-end Maintenance** ..... **49**

# Current Business

## Client Overview

The Georgia World Congress Center Authority (GWCCA) brings nearly 3 million visitors to downtown Atlanta every year, generating $1.8 billion in revenue for the city from hosted events and overnight stays at hotels. The Georgia World Congress Center (GWCC), managed by the GWCCA, is a for profit convention center and the third largest of its kind in the nation with a 3.9 million square feet campus. The center has held on average 300 events per year ranging from intimate community meetings to the Super Bowl Fan's Experience.

The scope of this project focuses on the GWCC site, which consists of 3 buildings (A, B, and C) that hold 1.5 million square feet of exhibit space. Four different types of rooms can be rented across all three buildings: Exhibit Halls, Ballrooms, Auditoriums, Meeting Rooms. The three buildings contain a total of 13 exhibit halls, 99 meeting rooms, 2 ballrooms, and 3 auditoriums.

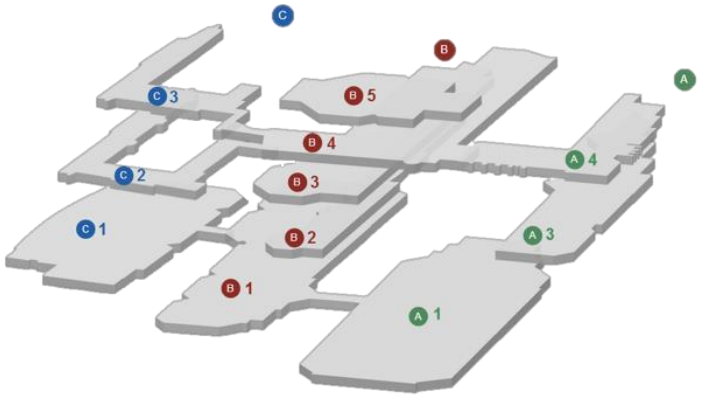



**Figure 1. GWCC Campus**

### System Description

The scope of this project is focused on the pre-negotiation process as shown in below.


**Figure 2. Process Flow**

Described below are the seven key process flow steps of the pre-event process:

1. The potential client sends a Request for Proposal (RFP) along with the inquiry.
2. The GWCCA confirms the availability of its space and other resources for the proposed event.
3. The center estimate the rental fees regarding the expected attendance and square footage.
4. Once the RFP is approved, a budget proposal is sent to the client and the negotiation process begins.
5. The client and the GWCCA sign the contract.

# Approach

---

### Goal and Methodology

The GWCCA aims to redesign their event valuation approach by collecting and analyzing data on an event-by-event basis. To help our client achieve this goal, our team analyzed the event expense data provided by different departments in the GWCCA and identified the opportunity for three methodologies: room assignment optimization, event-based cost prediction, and cost-based profit margin classification. We connect the three models and visualize the outputs of the models by designing a web app and a SQL database from scratch. The rooms chosen by the optimization model are used as inputs for the cost prediction model, which outputs a predicted event cost. The predicted cost is used by the profit margin classification which returns a suggested baseline price for the GWCCA to consider during negotiation.

### A. Room Assignment Optimization Model

*Motivation*

Currently, the GWCCA does not have room assignment guidelines, which would prevent several events from taking up several buildings that could instead be consolidated in one. For example, on a given day, all three buildings might be partially occupied by smaller events. Thus, when an incoming large-scale

event requests an entire building (known as “Under One Roof” to our client), the client would lose the business or force the event to be held across multiple buildings which drives down customer satisfaction. Our analysis shows that 25% of events the GWCCA was unable to book were due to failure of meeting “Under One Roof” requirement and 18% of past events were held in spread-out rooms. Furthermore, analysis demonstrates two rooms with similar size and functionality differ significantly when considering the weighted cost of an event depending on the location. To help our client reduce room costs and increase competitiveness during negotiation, our team designed an optimization model that selects rooms that historically are used for cheaper events for an incoming booking.

*Design Strategy*

The goal of this model is to select rooms that minimize costs and satisfy space requirements while consolidating rooms to be as close as possible. As for the inputs, the model stores the room cost, square feet, and coordinates of each room in the SQL database and requests minimum room size, number of exhibit halls, ballrooms, auditoriums, and meeting rooms from the GWCCA through the web app for each event. The objective of the model is to minimize the weighted room cost but and the total distance between selected rooms. The output of the model is a list of cost-efficient rooms. These are in turn converted to a count of exhibit halls, meeting rooms, and ballrooms that are used as additional inputs for the Cost Prediction Model. As after an event is booked the GWCCA cannot change the assigned rooms to an event afterwards, the model will not update assigned rooms as more events come in.

We took several steps to quantify the GWCCA’s business policies based on their experience into equations and constraints. We first estimated room cost per day by using past event expense data according to the square footage of each room[1]. Considering the assumption that the weighted room cost is constant across seasons and types of events, we took the average room cost using 2000 historical events.

Additionally, the GWCCA has space preferences for assigned rooms:

1. They should be in the same building, on the same floor, and next to each other.
2. If two buildings are used, buildings B and C are more favorable than A and B.

To satisfy the requirements, we designed 4-dimensional coordinates[2] (building, floor, x, y) for each room and assigned weights to each coordinate. We then constructed a distance matrix to calculate the distance

1 Appendix (B), 5

2 Appendix (B), 2

between selected rooms. To compromise the minimization of cost and distance, we include distance in the objective formula by multiplying it by a constant[3] we identified by testing 1000 past events.

Furthermore, we added constraints to make sure the model returns requested number of different types of rooms. To make sure all the rooms are large enough for the event, we included a constraint to make sure any room is larger than the required minimum squared footage. As some rooms cannot be selected together due to overlapping space, 6 conditional constraints were added. We added one more constraint to guarantee that only available rooms are selected on a specific date by syncing the model to the SQL database: using the date on the RFP, only available rooms on that date will be returned by the database.

In the end, the model contains 143 binary variables and 13 constraints [4]and is solved by Gurobi in Python. To validate the model, we tested the models with the 125 events from June to October of 2019. Our analysis validates that, client could select rooms that are 19% cheaper and decrease events held at spread-out layout by 13%.[5] We also provide the GWCCA the ability to modify selected rooms on the web app if event holder has special request.

## B. Cost Prediction Model

*Motivation*

The GWCCA's current cost budgeting tool only considers the square footage and number of attendees and does not improve as more data is collected. The specific formula is not available to our client as it was developed by a former employee. According to our analysis, the current tool predicts the cost with 61% deviation from true value. With the inaccurate cost, it is challenging for the GWCCA to decide which event to reject or accept as the profitability is uncertain. Thus, the GWCCA expressed the urgent need of a more accurate event-based cost prediction. To assist our client, we constructed a cost prediction model from scratch by starting from statistically identifying significant features impacting event cost.

*Design Strategy*

To improve event cost prediction, our team first identified significant features then designed a regression model that takes in the features and output a predicted cost for an upcoming event. Some of the features (the number of exhibit halls, meeting rooms, and ballrooms per building) are derived from the output of the Room Assignment Optimization Model.

3 Appendix (B), 3
4 Appendix (B), 4
5 Appendix (B), 6

After cleaning the data by mapping spreadsheets[6] provided by the GWCCA and generating extra features, 78 continuous and categorical features[7], which can potentially affect the cost performance, were obtained. To select the features with the most significant impact on the cost, feature selection was performed, which includes 5 correlation detection methods, 3 rank-based feature selection methods and 3 regression-based feature selection methods[8]. We considered both the results of technical analysis and business intuition to select features. For example, the public space in building A ('P-A') is shown as an average significant feature (ranked 19th.) based on selection methods; however, how much the public space is used in one building would not be available until the event occurs, so we decided to remove it. After removing all the "cheating variables," we finalized 13 features as following:

| Feature Name | Feature Source | Feature Explanation |
|---|---|---|
| sqftPerEvent | RFP | Square footage needed per event |
| orderedRentTotal | Bid and Revenue Calculator[9] | Total potential rental revenue can be generated per event (correlated with 'FB') |
| FB | Bid and Revenue Calculator | Minimum food and beverage revenue can be generated per event (correlated with 'orderedRentTotal') |
| Attendance | RFP | Expected attendance per event |
| totalRoomNights | RFP | Total number of nights the event holder request in hotel (No. of people * No. of nights) |
| contactTillStart | Feature Engineering[10] | The number of days in between the RFP submission date and the start date per event |
| eventLength | Feature Engineering | The number of days the event spans |

6 Appendix (A), 'Raw Data Provided by GWCCA'

7 Appendix (C), Section I

8 Appendix (C), Section I

9 Appendix (A), 'Raw Data Provided by GWCCA', 7

10 Appendix (C), Section I

| | | |
|---|---|---|
| E-A, E-B, E-C | Room Assignment Optimization Model | The number of exhibit halls assigned in building A, B, C separately per event |
| M-A, M-B, M-C | Room Assignment Optimization Model | The number of meeting rooms assigned in building A, B, C separately per event |

**Table 1. Features Selected for Cost Prediction Model**

We split the dataset which contains 529 events by using the most recent 80% of events in the for training[11] and the remaining 20% for the testing sets. We implemented regression algorithms and tuned hyper-parameters then used multiple metrics, such as $R^2$ and mean absolute error, to determine the best performing regression model[12].

The proposed model uses Machine Learning which could be considered as a "black box" that is difficult to interpret. In an effort to interpret the results of the model, we performed a sensitivity analysis to analyze how cost changes as each feature changes and all other features remaining constant. Through our analysis, we observed a nonlinear relationship between each feature and cost. For example, when the square footage of an event increases from 0.5 to 1.5 million, cost increases by 145%, but when square footage increases from 1.5 to 2.5 million, cost increases only by 56%. Similarly, we observe that cost increases significantly for events with over 5.5 million square feet.

After the model was constructed, a percent deviation[13] from the true expense for our model and the GWCCA's budget expense was computed to validate the model. By using the same 125 events we tested the Room Assignment Optimization Model with, on average, the GWCCA's expense budget had a 61% deviation from the true expense, while our proposed model has a 25% deviation from the true expense: our proposed model decreases the deviation by approximately 36% in the testing set and is able to predict expenses more accurately than the GWCCA's current tool.

11 Appendix (C), Section II, 'Randomized Grid Search with K – Fold Cross Validation'

12 Appendix (C), Section II

13 Appendix (C), Section II, 'Gradient Boosting', Figure 3.2.6 - 3.2.8

## C. Profit Margin Classification

*Motivation*

Currently, the GWCCA's pricing strategy is solely based on competitive pricing; they do not consider the cost, event type, and seasonality in demand when pricing for an incoming event. Consequently, current pricing method results in 6.4% of events to be held at a loss as the GWCCA charges for those events less than the actual costs.

*Design Strategy*

To improve the GWCCA's event profitability and decrease the number of events held at loss, our team proposes a classification tool to suggest an appropriate profit margin for each incoming event. The input of the classification is from the cost prediction model and the output is a recommended price for an upcoming event that our client can consider.

We analyzed the past events and identified patterns in profit margin for different types of events. Because 119 events used in the cost prediction model did not have the recorded revenue required for this analysis, our dataset for classification consists of 410 events. The type of an event was used to classify a cluster. The event types (exhibition, festivals, sports, etc.) that occurred infrequently in the past years were placed in the same cluster to generate the distribution of profit margins. In addition, some event types (charity and graduation, conference and games, film and award ceremonies) show similar profit margin distributions: shape, standard deviation, mean, and range as shown in Figure 3. Thus, we classified them in the same cluster.

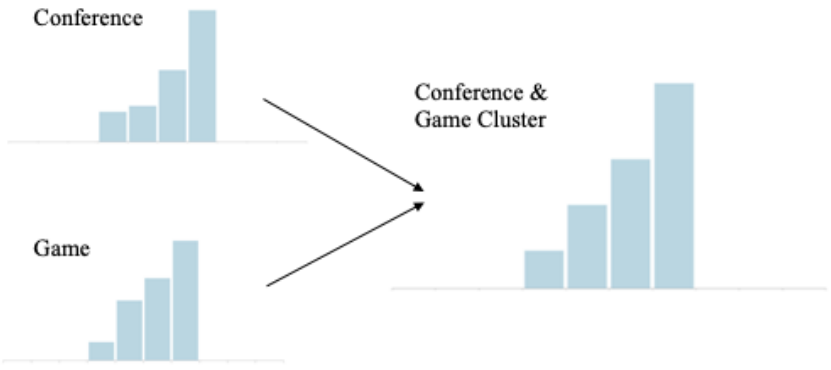



**Figure 3. Classification of Profit Margin Distribution - Conference & Game**

To further improve the classification and provide GWCCA a dynamic pricing strategy, we integrated demand seasonality. For each type cluster, we calculated the number of events that occurred for each month and divided into 3 different sub-clusters by demand level: low, medium, and high. Considering the profit margins for each month for each type cluster separately, we identify the months that fell under the 25th percentile of total events, months over 75th percentile of total events, low demand and medium demand respectively and the rest as high demand.

From the discussion with the GWCCA, the 50th percentile of the profit margin distribution was decided as a baseline price for medium demand sub-clusters, considering the likeliness to be accepted by event holders. For the low demand months, the 45th percentile of the distribution was used as a baseline price for future events so that the GWCCA can stay competitive in the market. For the high demand months, the 55th percentile of the distribution was used as a baseline price, so the GWCCA can generate more profit. Understanding that the GWCCA has years of experience in the convention industry, we provide the GWCCA the option to adjust and test different profit margins in the web app.

# Implementation

## Deliverable and Recommendations

To guarantee that GWCCA can easily access and interact with each of the three models, we packaged the three models into one web app. The web app was chosen as it's more flexible, user friendly, and accessible compared to desktop GUI and Excel Macros.[14] With the web app, the GWCCA can input event information, view and change the output of the room assignment model, view the predicted cost and it's 95% confidence interval, and interact with recommended baseline price.

To appropriately display all three models, the web app contains multiple pages as shown below.

| Page | Functionality |
|---|---|
| Home | Welcomes and orients user to web app |

14 Appendix (E)

| | |
|---|---|
| Create New Event | Enables input of new event information; begins running the three models |
| Room Output | Shows recommended rooms assigned to event; enables user to add/remove rooms |
| Cost Output | Displays confidence interval of cost of event and recommended profit margin for event |
| Search Events | Allows user to display details of booked events |
| Calendar | Shows booked events for any given month |
| Help | Details steps for maintaining web app |

**Table 2. Web App Outline**

Our team took the step to implement the web app. We first held a demo session to the key stakeholders then helped the GWCCA download required packages. We also held a training session with the head of the sales team and had a discussion about technical details with the IT team. As the GWCCA IT team is versed in Python, the rest of the implementation and maintenance can be handled with ease. As the GWCCA is already hosting a web app, our web app will be hosted on their server directly.

To maximize the potential of our tools, we identified some recommendations the GWCCA can consider in the future:

1. Accurately collect per-event expense.
2. Collect all inputs[15] on the Create New Event page when talking to a potential client.

## Risk and Mitigations

The optimization solver Gurobi involves a $10,000 capital cost. To mitigate such risk, our team prepared the code in an open-source solver PuLP. However, the solving time of Gurobi is significantly better than PuLP. After discussion, the GWCCA prefers Gurobi for the better solving performance and the potential of using it for other projects. The investment is ready to be made after a trial period.

15 Appendix (F)

The risk of delivering the models and a full-stack web app is the post-project maintenance. To make sure the GWCCA can use, maintain, and update each of our models as well as the web app, we prepared a detailed user manual [16]that even includes steps to take when each software is updated.

# Value & Validation

*Quantitative*

Using the web app, we simulated 125 events from June to October 2019 through the connecting three models. This simulation shows a 13% decrease in number of events held in spread out rooms, a 36% increase in accuracy of cost prediction, and $400,000 increase in profit[17]. We extrapolated the results across a year considering historical demand seasonality. The analysis shows an estimated $1.1 million annual profit increase for the GWCCA.

*Qualitative*

As the Room Assignment model outputs rooms that are close together, event attendee's satisfaction is expected to increase. The web app also provides a tool for the GWCCA to compare events, allowing them to make a more educated decision. Most importantly, the web app bridges the gap between the salespeople and the operations department by linking both of their expertise and increasing visibility of events' requirements. These improvements will allow "a top economic engine for the state of Georgia"[18] to increase their impact on the city of Atlanta by providing more public events and generating more money for our city.

16 Appendix (G)
17 Appendix (F)
18 References

# Appendix

## APPENDIX A: Data Collection and Cleaning

*Raw Data provided by GWCCA*

(Sensitive information is covered due to privacy protection)

1. Room type and area for all 3 buildings (Spreadsheet and pdf);

| | Colum | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | A | | B | | B/C | | C | |
| Row Labels | Sum of I | Sum of Gross | Sum o | Sum of Gross | Sum of | Sum of Gross | Sum of I | Sum of Gross |
| Ballroom/Aud | 1 | 15,689 | 4 | 33,674 | | | 3 | 25,748 |
| Exhibit | 3 | 340,000 | 5 | 607,500 | 1 | 107,670 | 4 | 418,500 |
| Mtg Rm | 30 | 69,938 | 48 | 116,284 | | | 26 | 52,253 |
| **Grand Total** | **34** | **425,627** | **57** | **757,458** | **1** | **107,670** | **33** | **496,501** |

Exhibit Halls

| Room or Area | Theater Capacity | 10x10 Booths Capacity | Banquet Capacity | Usable Area Sq. Ft. | Usable Area Sq. Meters | Ceiling Height |
|---|---|---|---|---|---|---|
| Exhibit Hall A1 | 15,113 | 800 | 8,940 | 149,000 | 13,848 | 30' |
| Exhibit Hall A2 | 10,244 | 490 | 5,620 | 86,000 | 7,992 | 30' |
| Exhibit Hall A3 | 9,570 | 574 | 5,340 | 105,000 | 9,758 | 30' |
| A1 – A3 | 34,927 | 1,864 | 19,900 | 340,000 | 31,598 | 30' |
| A1 – A2 | 25,357 | 1,290 | 14,560 | 235,000 | 21,840 | 30' |
| A2 – A3 | 19,814 | 1,064 | 10,960 | 191,000 | 17,750 | 30' |
| A/B Connector | | | | 30,153 | | |

Meeting Rooms*

| Room or Area | Theater Capacity | Classroom Capacity | Banquet Capacity | Usable Area Sq. Ft. | Usable Area Sq. Meters | Ceiling Height |
|---|---|---|---|---|---|---|
| A101 | 1,090 | 635 | 630 | 7,667 | 713 | 30' |
| A102 | 1,090 | 635 | 630 | 7,667 | 713 | 30' |
| A103 | 1,090 | 635 | 630 | 7,667 | 713 | 30' |
| A101 – A103 | 3,096 | 1,504 | 1,920 | 23,079 | 2,146 | 30' |

Figure 1.1.1 Sample Room & Building Information

2. Aggregated income statements from 13 departments (2018 – 2019);

April 2018 Departmental Income Statement _ FINAL — unfavorable

| | GL Account Description | PTD | PTD Bud | PTD Budget Variance | PTD Budget Var. % | | PTD (2) | Rolling Forecast |
|---|---|---|---|---|---|---|---|---|
| Grand Total | | | | | | | | |
| Accounting | | | | | | | | |
| Revenue (50) | | | | | | | | |
| | Salaries - Products and Services | | | | | | | |
| | Overtime - Products and Services | | | | | | | |
| | Temporary Help - Products and Services | | | | | | | |
| | F.I.C.A. - Products and Services | | | | | | | |
| | Retirement - Products and Services | | | | | | | |
| | Supplies Products and Services | | | | | | | |
| | Uniform/Laundry Products and Services | | | | | | | |
| | Advertising & Promotion Product Services | | | | | | | |
| | Dues, Subscriptions, & Fees - Product and Servi | | | | | | | |
| | Contractual -Other Products and Services | | | | | | | |
| | Contractual - Commercial Advertising | | | | | | | |
| | Contractual Sponsorship | | | | | | | |
| | Travel - Products and Services | | | | | | | |

Figure 1.1.2 Sample Income Statement

3. Account based expense records with event ID (2015 – 2019) from internal software (pulled by GWCCA);

**August FY18 Expense**

| GL Account Header | Amount | Event | End Date - JE |
|---|---|---|---|
| **Grand Total - Count: 952** | ######### | | |
| 2019-02 (Aug) (Closed) - Count: 952 | | | |
| Expense (70) - Count: 952 | | | |
| 018-65-106 Show Labor - Facility Ops | | Nike Tourname | 07/29/18 |
| 010-51-305 Part Time- Public Safety | | Atlanta Falcons | 08/17/18 |
| 018-65-329 4th of July | | 4th of July Cele | 07/04/18 |
| 018-65-329 4th of July | | 4th of July Cele | 07/04/18 |
| 018-65-106 Show Labor - Facility Ops | | 4th of July Cele | 07/04/18 |
| 018-65-107 Show Labor - Utility Services | | SEC Summer F | 07/15/18 |
| 018-65-329 4th of July | | 4th of July Cele | 07/04/18 |
| 018-65-106 Show Labor - Facility Ops | | V-103 Car and | 07/14/18 |
| 018-65-106 Show Labor - Facility Ops | | FCCLA National | 07/02/18 |

Figure 1.1.3 Sample Expense Raw Data

4. Account based revenue records with event ID (2015 - 2019) from internal software (pulled by the team);

| GL Account Header | Amount | Event | Year - Period | End Date - | Event ID - J |
|---|---|---|---|---|---|
| 005-44-615 Labor-Bldg. Eng | | ISTE 2014 | 2015-01 (Jul) (Closed) | 7/1/14 | 5102 |
| 005-44-617 Labor - Security | | ISTE 2014 | 2015-01 (Jul) (Closed) | 7/1/14 | 5102 |
| 005-44-617 Labor - Security | | ISTE 2014 | 2015-01 (Jul) (Closed) | 7/1/14 | 5102 |
| 005-44-618 Labor - Set Up | | ISTE 2014 | 2015-01 (Jul) (Closed) | 7/1/14 | 5102 |
| 005-44-501 Utility Services | | North American T | 2015-01 (Jul) (Closed) | 7/6/14 | 5254 |
| 305-44-301 Space Rental | | Park Market | 2015-01 (Jul) (Closed) | 6/27/14 | 8448 |
| 305-44-301 Space Rental | | Park Market | 2015-01 (Jul) (Closed) | 6/27/14 | 8448 |
| 305-44-301 Space Rental | | Park Market | 2015-01 (Jul) (Closed) | 6/27/14 | 8448 |
| 005-44-501 Utility Services | | North American T | 2015-01 (Jul) (Closed) | 7/6/14 | 5254 |
| 205-44-502 Parking | | Live Nation - On T | 2015-01 (Jul) (Closed) | 7/15/14 | 9152 |
| 205-44-501 Utility Services | | Live Nation - On T | 2015-01 (Jul) (Closed) | 7/15/14 | 9152 |

Figure 1.1.4 Sample Revenue Raw Data

5. Event list containing features such as event dates, expected attendance, total ordered rental, rooms, etc. with event ID (2016 - 2019);

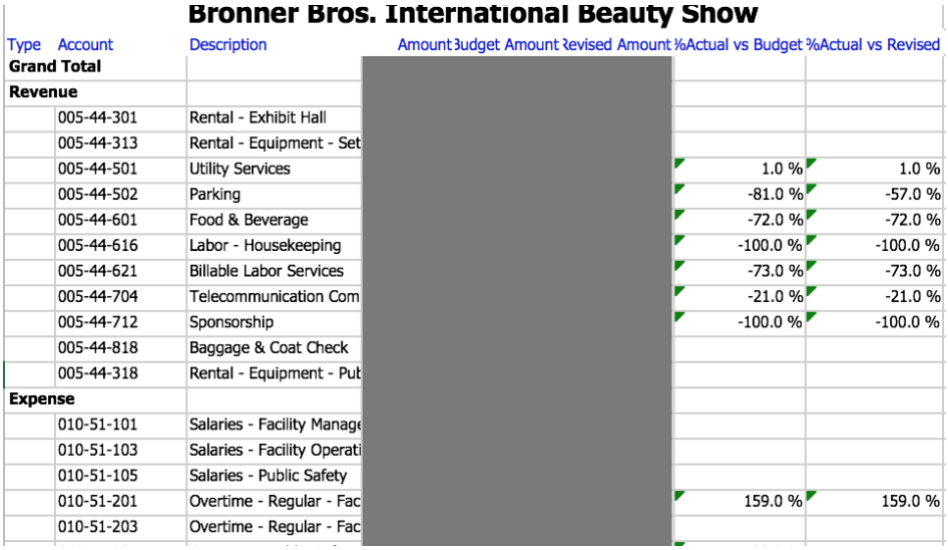
| Event | Description | Start Da | End Da | Type | Status |
|---|---|---|---|---|---|
| 20737 | SB- OPERATION | 01/02/19 | 02/21/19 | Game ( | Confirmed (56) |
| 20738 | SB- MEDIA --(U | 01/03/19 | 02/22/19 | Game ( | Confirmed (56) |
| 20739 | SB- STAFF TRAI | 01/03/19 | 02/22/19 | Game ( | Confirmed (56) |
| 20743 | SB- SUPER BOW | 01/03/19 | 02/22/19 | Game ( | Confirmed (56) |
| 20749 | SB- FRIDAY NIG | 01/03/19 | 02/22/19 | Game ( | Confirmed (56) |
| 20748 | SB- NFL TAILGA | 01/07/19 | 01/24/19 | Game ( | Confirmed (56) |
| 5237 | Progressive Insu | 01/10/19 | 01/13/19 | Public/C | Confirmed (56) |
| 21273 | Drone Commerc | 01/10/19 | 01/10/19 | Other ( | Confirmed (56) |
| 19271 | Herzing Univers | 01/11/19 | 01/11/19 | Graduat | Confirmed (56) |
| 19396 | Trane Reception | 01/14/19 | 01/14/19 | Conven | Confirmed (56) |
| 9925 | AHR Expo | 01/14/19 | 01/16/19 | Conven | Confirmed (56) |

| Forecast Attendan | Actual Attendanc | SQFT per Eve | Ordered Ren | Total service order rev |
|---|---|---|---|---|
| | | 13,871,372 | | |
| | | 2,401,158 | | |
| | | 313,780 | | |
| | | 11,646,301 | | |
| | | 497,090 | | |
| | | 4,857,600 | | |
| | | 10,138,118 | | |
| | | 94,390 | | |
| | | 41,368 | | |
| | | 25,722 | | |
| | | 21,246,321 | | |

Figure 1.1.5 Sample Event List Raw Data

6. Bid and Revenue Calculator: Built by a third party company, GWCC is not given the backend programming algorithm of the tool. The tool is used to create Ordered Rental Total and Food and Beverage revenue, including discount and waiver policies.

7. Expense and Revenue Budget event based data for about 30 events;

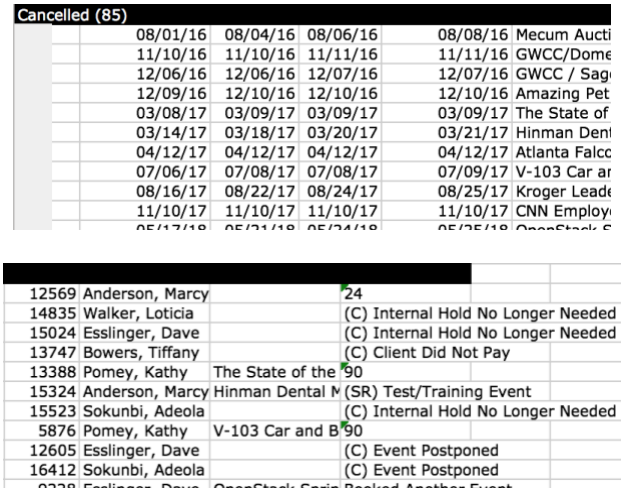


**Bronner Bros. International Beauty Show**

| Type | Account | Description | Amount | Budget Amount | Revised Amount | %Actual vs Budget | %Actual vs Revised |
|---|---|---|---|---|---|---|---|
| **Grand Total** | | | | | | | |
| **Revenue** | | | | | | | |
| | 005-44-301 | Rental - Exhibit Hall | | | | | |
| | 005-44-313 | Rental - Equipment - Set | | | | | |
| | 005-44-501 | Utility Services | | | | 1.0 % | 1.0 % |
| | 005-44-502 | Parking | | | | -81.0 % | -57.0 % |
| | 005-44-601 | Food & Beverage | | | | -72.0 % | -72.0 % |
| | 005-44-616 | Labor - Housekeeping | | | | -100.0 % | -100.0 % |
| | 005-44-621 | Billable Labor Services | | | | -73.0 % | -73.0 % |
| | 005-44-704 | Telecommunication Com | | | | -21.0 % | -21.0 % |
| | 005-44-712 | Sponsorship | | | | -100.0 % | -100.0 % |
| | 005-44-818 | Baggage & Coat Check | | | | | |
| | 005-44-318 | Rental - Equipment - Put | | | | | |
| **Expense** | | | | | | | |
| | 010-51-101 | Salaries - Facility Manage | | | | | |
| | 010-51-103 | Salaries - Facility Operati | | | | | |
| | 010-51-105 | Salaries - Public Safety | | | | | |
| | 010-51-201 | Overtime - Regular - Fac | | | | 159.0 % | 159.0 % |
| | 010-51-203 | Overtime - Regular - Fac | | | | | |

Figure 1.1.6 Sample Budget Information Raw Data

8. Cancelled and Lost Event Lists with event ID, event name, dates booked, salesman information and additional cancelled reason for some cancelled events;



**Cancelled (85)**

| | | | | | |
|---|---|---|---|---|---|
| | 08/01/16 | 08/04/16 | 08/06/16 | 08/08/16 | Mecum Aucti |
| | 11/10/16 | 11/10/16 | 11/11/16 | 11/11/16 | GWCC/Dome |
| | 12/06/16 | 12/06/16 | 12/07/16 | 12/07/16 | GWCC / Sag |
| | 12/09/16 | 12/10/16 | 12/10/16 | 12/10/16 | Amazing Pet |
| | 03/08/17 | 03/09/17 | 03/09/17 | 03/09/17 | The State of |
| | 03/14/17 | 03/18/17 | 03/20/17 | 03/21/17 | Hinman Dent |
| | 04/12/17 | 04/12/17 | 04/12/17 | 04/12/17 | Atlanta Falcc |
| | 07/06/17 | 07/08/17 | 07/08/17 | 07/09/17 | V-103 Car ar |
| | 08/16/17 | 08/22/17 | 08/24/17 | 08/25/17 | Kroger Leade |
| | 11/10/17 | 11/10/17 | 11/10/17 | 11/10/17 | CNN Employ |

| | | | |
|---|---|---|---|
| 12569 | Anderson, Marcy | | 24 |
| 14835 | Walker, Loticia | | (C) Internal Hold No Longer Needed |
| 15024 | Esslinger, Dave | | (C) Internal Hold No Longer Needed |
| 13747 | Bowers, Tiffany | | (C) Client Did Not Pay |
| 13388 | Pomey, Kathy | The State of the | 90 |
| 15324 | Anderson, Marcy | Hinman Dental M | (SR) Test/Training Event |
| 15523 | Sokunbi, Adeola | | (C) Internal Hold No Longer Needed |
| 5876 | Pomey, Kathy | V-103 Car and B | 90 |
| 12605 | Esslinger, Dave | | (C) Event Postponed |
| 16412 | Sokunbi, Adeola | | (C) Event Postponed |

Figure 1.1.7 Sample Cancelled Events Raw Data

9. Additional Event list including specific room and book day detail, with event ID (2016 - 2019);

| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| Event | Descriptio | Status | Start Da | End Da | Type | Booked Spaces | Firm (B |
| 6179 | Passion Confer | Confirmed (56) | 01/01/15 | 01/03/15 | Conventic | Exhibit Hall A2, Exhibit Hall B2, | 06/06/14 |
| 5045 | AutoTrader.co | Confirmed (56) | 01/06/15 | 01/08/15 | Conferenc | Building A Registration Hall, Ex | 04/20/09 |
| 9062 | Girl Scouts of | Confirmed (56) | 01/09/15 | 01/11/15 | Meeting ( | Exhibit Hall A2, Meeting Room | 09/30/14 |
| 7774 | 2015 Georgia | Confirmed (56) | 01/10/15 | 01/10/15 | Co-Produc | Building A Registration Hall, Ex | 04/15/14 |
| 8130 | 2015 Internati | Confirmed (56) | 01/10/15 | 01/11/15 | Meeting ( | Georgia Ballroom 1-3 | 03/18/14 |
| 5883 | Monster Jam | Confirmed (56) | 01/10/15 | 01/10/15 | Competiti | Stadium, West Plaza | 10/17/14 |
| 8047 | Novo Nordisk | Confirmed (56) | 01/11/15 | 01/14/15 | Conferenc | Executive Boardroom, Exhibit H | 09/18/14 |
| 5510 | Eggs & Issues | Confirmed (56) | 01/13/15 | 01/13/15 | Breakfast | Building B Registration Hall, Me | 10/21/14 |
| 10099 | Jeff Zucker CN | Confirmed (56) | 01/13/15 | 01/14/15 | Meeting ( | Sidney Marcus Auditorium | 12/29/14 |

Figure 1.1.8 Sample Additional Event Room Records Raw Data

*Data Cleaning*

In general, we identified the uniqueness of 'Event ID' and used it as a unique ID for mapping.

1. Expense data cleaning
    a. Calculate aggregated expense from expense raw data, event based
    b. Remove event entries with overall expense equals to 0 or negative
2. Revenue data cleaning
    a. Calculate aggregated revenue from revenue raw data, event based
    b. Remove event entries with overall revenue equals to 0 or negative
3. Combined data cleaning
    a. Remove event entries with no feature information, from event list record
    b. Remove outliers whose expense far exceeds revenue (generating -200% profit margin)
    c. Remove event entries whose 'SQFT Per Event' feature has 0 or negative values
    d. Rename 'Type' feature from English to self-defined tag, such as 'CONV' for 'convention'

| Category | Code |
|---|---|
| Convention | CONV |
| Conference | CONF |
| Competition | COMP |
| Meeting | MEET |
| Parking | P |
| Meal | ME |
| Training | TRAI |
| Charity | CHRTY |
| Graduation | GRAD |
| Banquet/Reception | REC |

Figure 1.1.9 Sample Event Type Mapping

    e. Rename 'Room' feature from English to self-defined tag, such as 'M-A1' for Meeting room in Building A, small area occupation (size is generated based on square footage distribution for each type of rooms)

| index | Space Name | Category |
|---|---|---|
| 0 | Exhibit Hall A2 | E-A1 |
| 1 | Exhibit Hall B2 | E-B1 |
| 2 | Meeting Room A307 | M-A1 |
| 3 | Meeting Room A402 | M-A1 |
| 4 | Meeting Room A403 | M-A1 |
| 5 | Meeting Room A404 | M-A2 |
| 6 | Meeting Room A411-A412 | M-A5, M-A5 |
| 7 | Meeting Room B208 | M-B3 |
| 8 | Meeting Room B302-B305 | M-B3, M-B4, M-B4, M-B3 |
| 9 | Meeting Room B312-B314 | M-B4, M-B5, M-B4 |
| 10 | Meeting Room B405-B407 | M-B4, M-B5, M-B4 |
| 11 | Building A Registration Hall | P-A |
| 12 | Exhibit Hall A1 | E-A3 |
| 13 | Exhibit Hall A3 | E-A2 |

Figure 1.1.10 Sample Room Grouping Results

| | Name Convention | | |
|---|---|---|---|
| eg: | Exhibit Hall A2 | | E-A1 |
| | Meeting Room B208 | | M-B2 |
| | Building  A Registration Hall | | P-A |
| | Stadium | | O |
| | Meeting Room 1 | | U |
| | Sydney Marcus Auditorium | | BA-A |

Figure 1.1.11 Sample Room Tag Name Convention

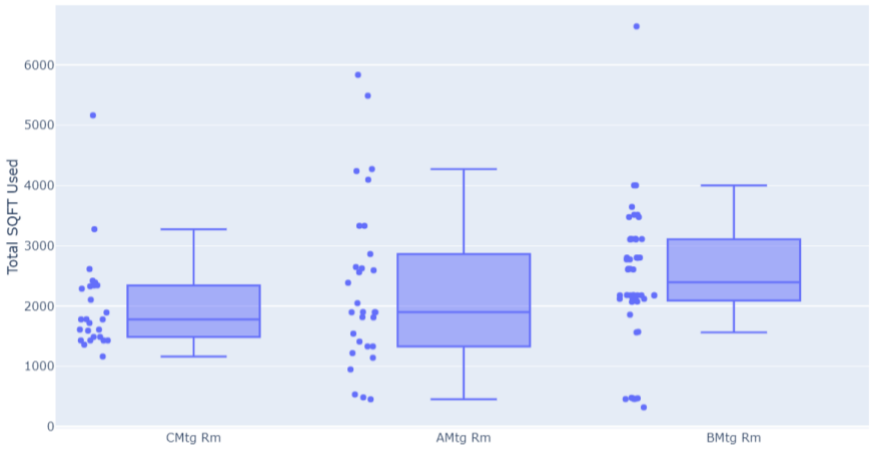



Figure 1.1.12 Sample Room SQFT Distribution

## APPENDIX B: Room Assignment Model

*1. Room Cost Estimation*

The following process was used to calculate the expense per day of each room. Linearity between expense and square footage was a main assumption. For each event, each room was given a weight according to its square footage. The total expense of an event was then divided according to these weights to each room involved in the event and the number of days of the event. This calculation resulted in room

cost per day for a given event. By averaging this value across all events, a room cost per day was calculated to be used in the optimization model.

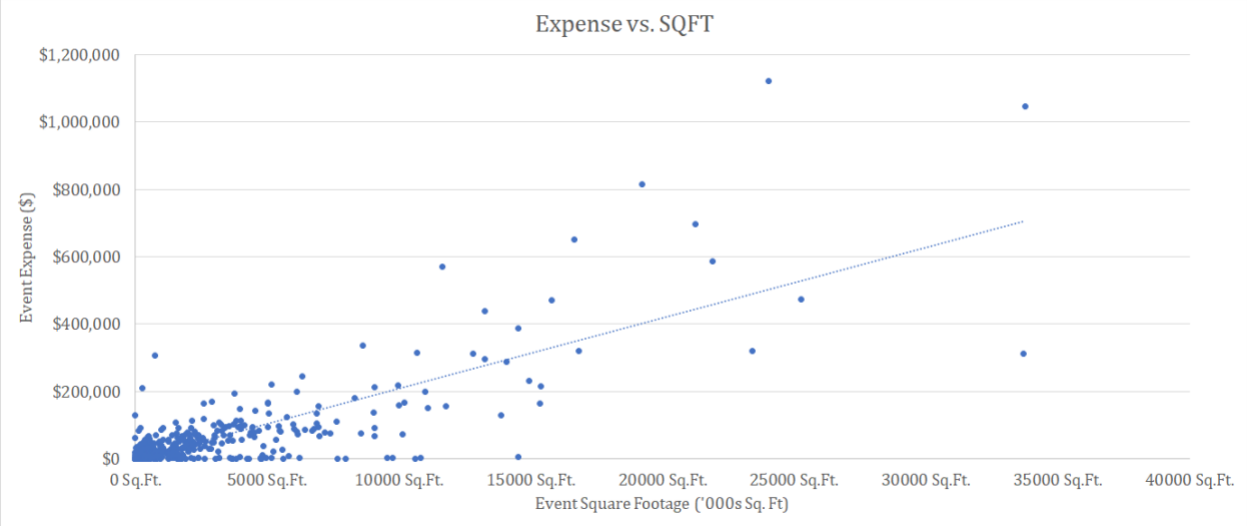



Figure 2.1.1 Relationship Between Expense and Square Footage of Event Rooms

*2. 4-Dimensional Coordinate Design*

To quantify the distance among the rooms, a 4-dimensional coordinate is assigned to each room in following convention: (Building, Floor, X, Y) in which the origin for X and Y is at the left bottom of each floor plan. Additionally, each floor and buildings are assigned with a weight relating to their distance. The weighted distance between Building C and B is smaller than that of Building A and B to reflect that Building B and C are preferred to Building A and B combo if two buildings have to be chosen at the same time. On each floor, X and Y coordinates were assigned based on the relative distance of each room to the origin. For example, the coordinate of Room A301[19] is (A,3,1,2). A difference vector will be calculated as the average distance of each room to a calculated centroid. The length of the difference vector[20] will be used in the model.

| | |
|---|---|
| Building A | 0 |
| Building B | 100 |
| Buidling C | 130 |
| Floor 1 | 10 |
| Floor 2 | 20 |
| Floor 3 | 30 |
| Floor 4 | 40 |
| Floor 5 | 50 |

Figure 2.1.2 Weighted Distance per Building and Floor

19 Figure 2.1.3

20 Equation 2.1

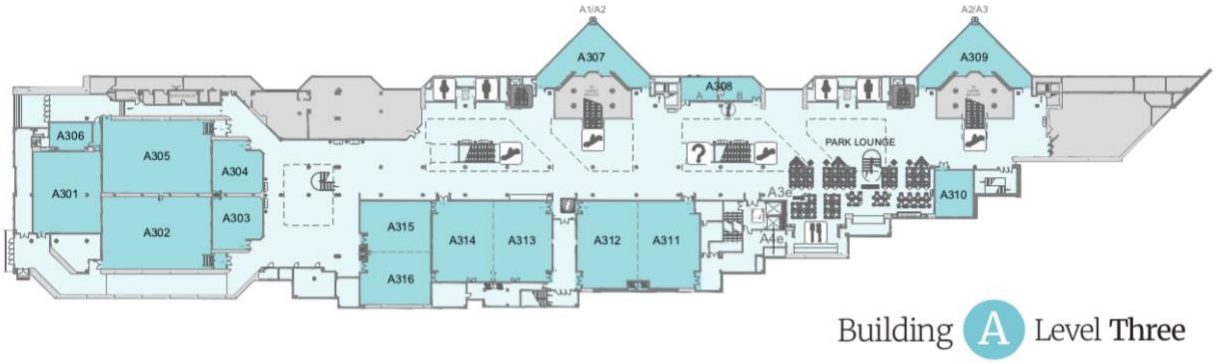



Figure 2.1.3 Building A Floor Plan

$$\|\mathbf{x}\| := \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Equation 2.1.4 Length of the Difference Vector

*3. Distance Multiplier*

To find the most reasonable constant relating distance between rooms to cost in the objective function. We tested different numbers with 125 events from June to October 2019 till we found the one that returns the closest rooms in all the cases. In the end, 7 is identified to be the constant.

*4. Formulation*

**Variables:**
$x_i$: {1 if a room is used, 0 otherwise, ∀ i ∈ {1, ..., 143}
$E$: indicates whether an exhibit hall is needed, 1 or 0
$A$: indicates whether an auditorium is needed, 1 or 0
$B$: indicates whether a ballroom is needed, 1 or 0

*Dummy Variables*
$m$: 7; a multiplier relating distance to cost
$d$: the average distance of rooms selected (d = centroid (pi for all i that's selected by the model)

**Data:**
$E$: number of exhibit halls requested by client
$A$: number of auditroiums requested by client
$B$: number of ballrooms requested by client
$G$: number of meeting rooms requested by client
$p_i$: coordinate of room i, ∀ i ∈ {1,…, 143}
$c_i$: cost per day of maintaining room i ∀ i ∈ {1,…, 143}
$s_i$: square foot of room i ∀ i ∈ {1,…, 143}
$SR$: *minimum* square footage of room

Let $e$ be the set of Exhibit Halls where i ∈ { 0,1,2,40,41,42,43,44,106,107, 108, 109, 143, 23, 24,87}
Let $a$ be the set of Auditoriums where i ∈{ 9, 122, 123}
Let $b$ be the set of Ballrooms where i ∈ { 102,103,104,105,137,138,139}
Let g be the set of Meeting rooms where i ∈ {*Not in e or a or b*}
Let O be the set of occupied rooms for the selected dates where the indexes change as the selected dates change.

**Objective:** $\min \sum c_i x_i + md$ [Minimize total cost maintaining rooms.]

**s.t**

$\sum_{i \subset O} x_i = 0$ [Cannot book the room that's already booked]

$\sum_{i \subset e} x_i >= E$ [Make sure the right number of exhibit halls is provided]

$\sum_{i \subset a} x_i >= A$ [Make sure the right number of auditoriums is provided]

$\sum_{i \subset b} x_i >= B$ [Make sure the right number of ballrooms is provided]

$\sum_{i \subset g} x_i >= G$ [Make sure the right number of meeting rooms is provided]

$\sum_{i \subset U} x_i >= SR$ [Make sure selected rooms are larger than minimal requirement]

$x_2 * (x_4 + x_5 + x_6) == 0$

$x_{40} * (x_{45} + x_{46} + x_{47}) == 0$

$x_{106} * (x_{110} + x_{111} + x_{112} + x_{113}) == 0$

$x_{142} * (x_{13} + x_{14}) == 0$

$x_{78} * (x_{79} + x_{80}) == 0$

$x_{94} * (x_{95} + x_{96}) == 0$

} Ensure no overlapping between rooms

Figure 2.1.5 - 2.1.6 Model Formulation

```
import pandas as pd
import numpy as np
import math
from gurobipy import Model,GRB,quicksum

data = pd.read_csv('opt1.csv', index_col=0, header=0)
roomsIndex = list(range(0,len(data.iloc[:,0])))

m = Model('Rooms')
m.setParam('MIPGap',0.5)
x = m.addVars(roomsIndex,name="x",vtype=GRB.BINARY)
d = m.addVar(0, name="d", vtype=GRB.CONTINUOUS)
DearDaniel = 7

#S = data.iloc[2,10]
E = data.iloc[3,10]
A = data.iloc[4,10]
B = data.iloc[5,10]
G = data.iloc[6,10]
SR = data.iloc[7,10]
#Available rooms
Arooms = data.loc[data['Occupied'] == 0]
AroomsIndex = data.loc[data['Occupied'] == 0].index.values.tolist()
Orooms = data.loc[data['Occupied'] == 1].index.values.tolist()
#Available rooms
#dictArooms = dict(roomsIndex,Arooms)
#print(Arooms)
#print(type(Arooms))

e = [0,1,2,40,41,42,43,44,106,107, 108, 109, 143, 23, 24,87]
a = [39, 122, 123]
b = [102,103,104,105,137,138,139]
g = np.setdiff1d(roomsIndex,e+a+b)

m.setObjective((quicksum(x[i] * data.iloc[i,2] for i in roomsIndex)+ DearDaniel*d) ,GRB.MINIMIZE)
#space requirment
#m.addConstr(quicksum(x[i] * data.iloc[i,1] for i in roomsIndex)>= S)
#no occupied room
m.addConstr(sum(x[i] for i in Orooms) == 0)
#Exhibit hall is provided while requested
m.addConstr(sum(x[i] for i in e)>= E)
```

```
m.addConstr(sum(x[i] for i in a)>= A)
#Ballroom is provided while requested
m.addConstr(sum(x[i] for i in b)>= B)
#Meeting Rooms
m.addConstr(sum(x[i] for i in g)>= G)
#if E is selected, Meeting room in that can't be selected
m.addConstr((x[2]*(x[4]+x[5]+x[3]))==0)
m.addConstr((x[40]*(x[46]+x[47]+x[45]))==0)
m.addConstr((x[106]*(x[112]+x[113]+x[110]+x[111]))==0)
m.addConstr(x[142]*(x[13]+x[14])==0)
m.addConstr(x[78]*(x[79]+x[80])==0)
m.addConstr(x[94]*(x[95]+x[96])==0)
#Min SQFT
m.addConstrs(x[i] == 0 for i in roomsIndex if data.iloc[i,1] < SR)
#distance stuff
pRooms = [(Arooms.iloc[i,3],Arooms.iloc[i,4],Arooms.iloc[i,5],Arooms.iloc[i,6])
for i in range(len(AroomsIndex))]

Sum = 0
distMat = np.empty([len(Arooms), len(Arooms)])
i = 0
while i < len(Arooms):
    room1 = pRooms[i]
    j = i
    while j < len(Arooms):
        room2 = pRooms[j]
        adiff = [((room1[k] - room2[k])**2) for k in range(4)]
        dist = math.sqrt(sum(adiff))
        distMat[i][j] = dist
        distMat[j][i] = dist
        Sum += dist
        j+=1
    i+=1

#distance calculation
m.addConstr(d == sum([((x[i]*x[j]) * distMat[i][j]) for i in range(len(AroomsIndex))
    for j in range(len(AroomsIndex))]))
m.optimize()
```

Figure 2.1.7 - 2.1.8 Model Code

*5. SQL Database*

```
CREATE TABLE user(
    employeeId TEXT NOT NULL,
    empPassword TEXT NOT NULL
);

CREATE TABLE ROOM(
    RoomID          INTEGER PRIMARY KEY,
    Name            TEXT NOT NULL,
    Sqft            INTEGER NOT NULL,
    Cost            INTEGER NOT NULL,
    Building        INTEGER NOT NULL,
    Floor           INTEGER NOT NULL,
    X               INTEGER NOT NULL,
    Y               INTEGER NOT NULL,
);

CREATE TABLE BOOKED(
    RoomID          INTEGER NOT NULL,
    Name            TEXT NOT NULL,
    DateIN          INTEGER NOT NULL,
    DateOut         INTEGER NOT NULL,
    PRIMARY KEY (RoomID,DateIn,DateOut),
    FOREIGN KEY(RoomID) REFERENCES ROOM(RoomID)
        ON DELETE CASCADE ON UPDATE CASCADE
);



.separator ,
.import loginCredentials.csv user
.import Rooms.csv ROOM
.import OccupiedRoomsData.csv BOOKED
```

Figure 2.1.9 Database Structure

*6. Validation Calculation*

| | | | | | | |
|---|---|---|---|---|---|---|
| 20660 | 3044.64 | 1830.152 | Exhibit Hall C1, Exl | ['Exhibit Hall B3 | 0 | 0 |
| 20688 | 2537.11 | 1636.406 | Exhibit Hall A2, M | ['Exhibit Hall B1 | 0 | 0 |
| 20915 | 4146.53 | 284.4314 | Georgia Ballroom : | ['Thomas Murp | 0 | 0 |
| 21014 | 1498.33 | 284.1108 | ['Executive Boardr | ['Exhibit Hall M | 0 | 0 |
| 21096 | 1537.4 | 877.4941 | Georgia Ballroom : | ['Thomas Murp | 0 | 0 |
| 21144 | 1287.7 | 105.0772 | Meeting Room B4( | ['Meeting Roon | 0 | 0 |
| 21162 | 6612.76 | 6322.704 | Exhibit Hall C2, Exl | ['Exhibit Hall C1 | 0 | 0 |
| 21182 | 930.93 | 7.557797 | Meeting Room A4( | ['Meeting Roon | 0 | 0 |
| 21237 | 125.77 | 76.98507 | ['Meeting Room B. | ['Meeting Roon | 0 | 0 |
| 21416 | 3185.48 | 426.5513 | ['Exhibit Hall A3', ' | ['Building B Reg | 0 | 1 |
| 21429 | 2097.94 | 19.6868 | Meeting Room C2( | ['Meeting Roon | 0 | 0 |
| 22208 | 587.83 | 185.8957 | Meeting Room A4 | ['Exhibit Hall M | 0 | 0 |
| 22215 | 587.83 | 185.8957 | ['Meeting Room A. | ['Exhibit Hall M | 0 | 0 |
| 22400 | 15441.31 | 16894.86 | ['Exhibit Hall A1', ' | ['Exhibit Hall A1 | 1 | 0 |
| 22402 | 20446.67 | 1086.889 | Exhibit Hall A1, Exl | ['Building B Reg | 1 | 0 |
| 22700 | 1964.51 | 865.9869 | Georgia Ballroom : | ['Thomas Murp | 0 | 0 |
| 22849 | 1213.83 | 6.655077 | Meeting Room B4: | ['Meeting Roon | 0 | 0 |
| 22975 | 201.69 | 77.69516 | Meeting Room B4( | ['Meeting Roon | 0 | 0 |
| 23024 | 3601.41 | 2206.179 | Exhibit Hall A3, Exl | ['Exhibit Hall C3 | 0 | 0 |
| 23195 | 125.77 | 105.0772 | Meeting Room B4( | ['Meeting Roon | 0 | 0 |
| Totals: | 259794.4 | 208361.4 | | | 8 | 7 |
| % Change: | | -20% | | | | -12.50% |

Figure 2.1.10 Model Validation Calculation

## APPENDIX C: Cost Prediction Model

### Section I Feature Introduction and Feature Elimination

*Feature Engineering*

The motivation of performing feature engineering is that in order to have accurate models, having representative features is essential. We need to start with as many features as possible, and perform feature dimension reduction to find the ones which are most significant/have greatest impact on response variable, which in our model is the cost. Here is a list with all the additional potential influential features we created with feature engineering:

| Name | Type | Explanation | Motivation |
|---|---|---|---|
| Attendance | Discrete | Taking the maximum of expected attendance and actual attendance | 38.93% events do not have actual attendance record for the scale of events is too large, so we take a maximum of the expected and actual to give conservative results |
| contactTillStart | Discrete | Taking the difference between the RFP submission date and the start date of event | The difference between the submission and start dates implies the urgency of events and could affect both costs and revenue |
| bookingTillStart | Discrete | Taking the difference between the book entered date and the start date of event | The difference between the book information entered and start dates implies how much GWCCA is interested in one event and could affect both costs and revenue |
| eventDuration | Discrete | Taking the difference between event end date and start date | Original features are start and end dates; we use this length to reduce 2 features into 1, also need another feature to indicate seasonal impact |
| startDoW | Discrete | Days it takes for one event to move in | During move-in process, costs can be different from both no event days and with event days |
| endDow | Discrete | Days it takes for one event to move out | During move-out process, costs can be different from both no event days and with event days |
| TotalEventLength | Discrete | Taking the difference between event move out date and move in date | Taking the above 3 features into account, but we were not sure if individual feature would be more significant than the aggregated, correlations were verified in next section |
| isWeekend | Binary | If the event dates include weekends | Take care of potential impact of extra payment of labor and other facility on weekends |
| isHoliday | Binary | If the event dates include special holidays | Take care of potential impact of extra payment of labor and other facility during holidays |

| | | | |
|---|---|---|---|
| Recur(Y/N) | Binary | If the event is a recurring event, which means same event ID appears more than once | Recurring events usually have long term contract with GWCCA, so the pricing can be lower by present standard, which can potentially impact the revenue |
| Frequency | Discrete | If the event is recurring, the number of occurrences based on all historical records | High frequency events are sometimes offered with better structured labor, because the request for labor is likely to be consistent, and easy to predict |

*Feature Selection*

As described in the report, over 78 features to start with, compared with less than 600 events, are too many, which can make our model suffer from overfitting or easily be affected by outliers. In order to avoid bad performance, because we cannot get more historical events, we need to select features which have the most significant impact on expense performance.

3 rank based feature selection models were implemented: F-Test regression selection ranking, Mutual Information regression selection ranking, Gradient Boosting feature importance ranking. One of the drawbacks of F-Test and Gradient Boosting ranking is that highly correlated features could jeopardize the ranking, for the F-Test gives higher score for highly correlated features and low score for less correlated features. Gradient Boosting tends to give low feature importance to correlated features. In order to avoid having highly correlated features, we performed 4 pair-wise correlation tests and 1 overall correlation test on 1- n correlation for features and we remove the ones jeopardize the rankings most.

1. Continuous vs. Continuous variable correlation check
    a. Pearson correlation: Checks only linear relationship, have strong assumptions on residuals (equal variance and no pattern/random distribution of variance).
    b. Spearman correlation: Checks higher order correlations, no assumptions required. But usually used to check for ranked/ordinal variables.

Because each method has drawback, we performed both and select the ones both indicated a strong correlation (with a correlation coefficient > 0.7).

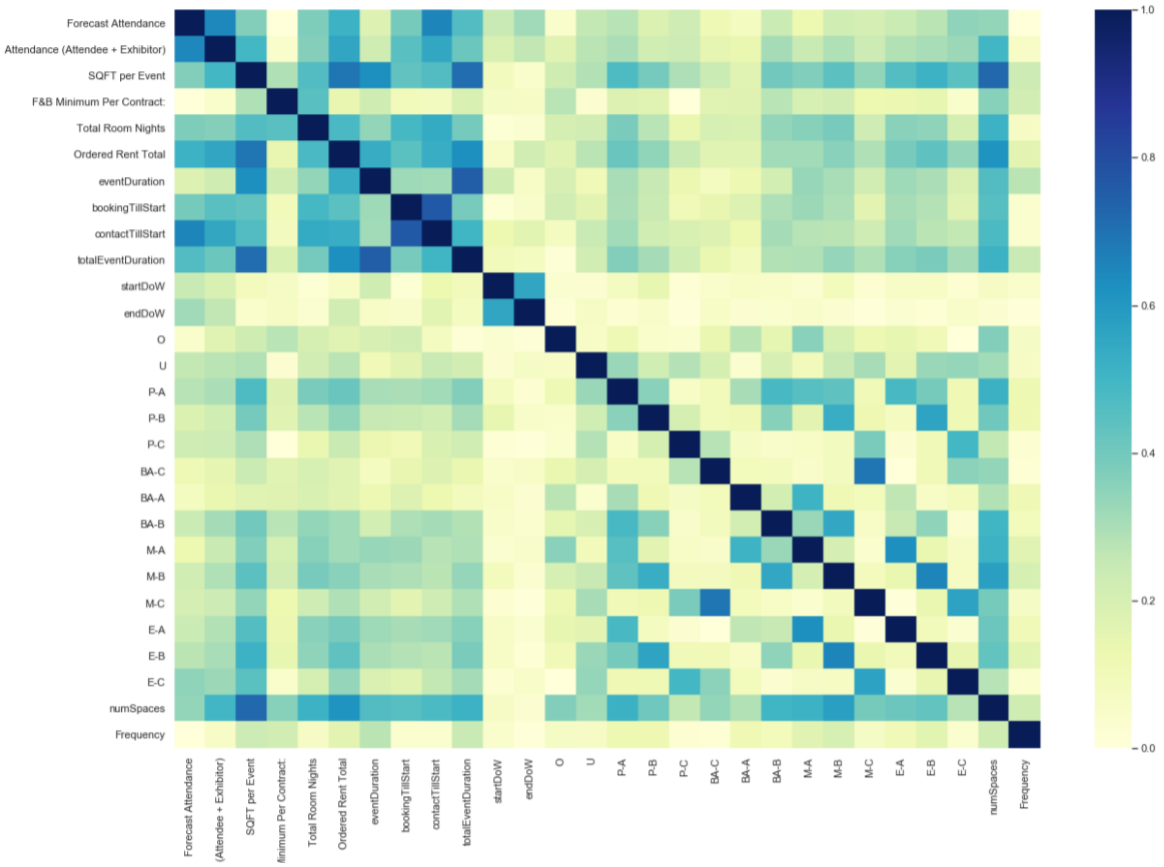



Figure 3.1.1 Sample Spearman Correlation Test Output

| | Var1 | Var2 | corr |
|---|---|---|---|
| 0 | bookingTillStart | contactTillStart | 0.762132 |
| 1 | eventDuration | totalEventDuration | 0.750418 |
| 2 | SQFT per Event | numSpaces | 0.724498 |
| 3 | SQFT per Event | totalEventDuration | 0.713280 |
| 4 | SQFT per Event | Ordered Rent Total | 0.691848 |

Figure 3.1.2 Sample High Correlated Continuous Features

2. Categorical vs. Categorical variable correlation and Categorical vs. Continuous variable correlation check

   a. Cramer's V: Check categorical and categorical variables correlation, by giving correlation coefficients in [0,1], no pre-assumptions for residual distribution.

   b. Point Biserial Correlation: Check continuous and categorical variables correlation, assumptions about normality and homoscedasticity are needed.

Both types of correlations are not showing results with correlation coefficients greater than or equal to 0.7.

3. Variance Inflation Factor (VIF)

VIF captures the multicollinearity of an ordinary least squares (OLS) regression, which takes the correlations among more than pairwise features into consideration. Thus, it is a good measurement for overall correlation, also it takes in only continuous variables. The following is the result features after removing all the features whose VIF's $\geq 10$.

| | VIF Factor | features |
|---|---|---|
| 1 | 6.755382 | Ordered Rent Total |
| 2 | 6.188155 | SQFT per Event |
| 3 | 6.071802 | endDoW |
| 4 | 5.861711 | startDoW |
| 6 | 3.154984 | Total Room Nights |
| 5 | 2.630981 | contactTillStart |
| 7 | 1.680537 | Attendance |
| 8 | 1.405194 | F&B Minimum Per Contract: |
| 0 | 1.196030 | eventDuration |
| 9 | 1.066011 | Frequency |

Figure 3.1.3 Acceptable not highly correlated features (VIF < 10)

(iv). Rank based: F-Test regression selection ranking

In order to compare the significance of each feature, F-Test tests one feature at a time, and start with a $Y_0 = c$, as $c$ is a constant. From the existing feature pool, F-Test takes one feature in an iteration to construct a model $Y_1 = \beta^1 \cdot X_1 + c$. By setting a hypothesis test:

$$H_0: \beta^1 = 0$$

$$H_\alpha: \beta^1 \neq 0$$

Equation 3.1

to test if the chosen feature $X_1$ is significant enough. If we set significance level $\alpha = 0.1$, then any $p - value < \alpha$ will lead to reject the null hypothesis, so we could conclude the feature being tested is significant. Based on the above testing, a ranking is given to each feature, and here is part of the rank list:

| feature | p-value | rank |
|---|---|---|
| SQFT per Event | 3.898430e-71 | 1 |
| Ordered Rent Total | 5.237793e-55 | 2 |
| Total Room Nights | 1.021227e-38 | 3 |
| E-B | 2.606112e-31 | 4 |
| M-B | 3.086681e-24 | 5 |
| E-C | 3.550820e-23 | 6 |
| M-C | 1.789230e-21 | 7 |
| Type_CONV | 3.506347e-20 | 8 |
| E-A | 6.366652e-20 | 9 |
| M-A | 4.316549e-19 | 10 |

Figure 3.1.4 Sample F-Test Rank List

The drawback of this method is that it only captures linear relationship between the feature and response, and it does not perform well for correlated terms, but we already removed high correlated features, so we would not suffer from the drawbacks.

4. Rank based: Mutual information regression

Mutual information (MI) of two variables measures the mutual dependence between the two variables. MI quantifies the amount of information contained in one variable by observing the other variable. The following equation is used to calculate MI, where H represents entropy, which is describing the disorder of uncertainty of one feature.

$$I(X,Y) = H(X,Y) - H(Y) - H(Y|X)$$

Equation 3.2

In our model, in each iteration, MI measures how much information we can get about cost by observing only one feature at a time. The output score is a 0 only if the two variables are independent, and higher values mean higher dependency.

Because the algorithm uses K-nearest-neighbor distance to measure the entropy, so a parameter ‘n_neighbors’ needs to be determined for how many data points we want the algorithm to consider within a fixed distance. The parameter cannot be too high, otherwise it increases the bias. The following chart showed some of the hyper-parameters we tested and we used average ranking as a reference.

| feature | scores3 | scores6 | scores7 | scores8 | scores9 | scores10 | scores11 | scores12 | scores13 | average | rank |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ordered Rent Total | 0.415721 | 0.338347 | 0.342176 | 0.333600 | 0.333281 | 0.325853 | 0.327433 | 0.326061 | 0.323883 | 0.343471 | 1 |
| Attendance | 0.361958 | 0.363753 | 0.349184 | 0.336770 | 0.319617 | 0.308911 | 0.303896 | 0.300808 | 0.299275 | 0.335968 | 2 |
| SQFT per Event | 0.362452 | 0.316820 | 0.317881 | 0.315328 | 0.297312 | 0.291571 | 0.292268 | 0.291747 | 0.291790 | 0.311242 | 3 |
| contactTillStart | 0.259887 | 0.295453 | 0.287852 | 0.278407 | 0.277297 | 0.271906 | 0.269137 | 0.262723 | 0.259777 | 0.275241 | 4 |
| Total Room Nights | 0.197230 | 0.164110 | 0.157880 | 0.155151 | 0.149958 | 0.148181 | 0.155639 | 0.157102 | 0.158086 | 0.162999 | 5 |
| eventDuration | 0.188273 | 0.156589 | 0.142320 | 0.142506 | 0.143096 | 0.146587 | 0.144128 | 0.140783 | 0.131266 | 0.153003 | 6 |
| Type_CONV | 0.119814 | 0.118799 | 0.113967 | 0.111011 | 0.105581 | 0.106279 | 0.106995 | 0.107242 | 0.106727 | 0.113240 | 7 |
| Frequency | 0.138164 | 0.110087 | 0.094067 | 0.091687 | 0.092198 | 0.085567 | 0.086319 | 0.078903 | 0.075288 | 0.097812 | 8 |
| M-B | 0.103137 | 0.089560 | 0.084904 | 0.083611 | 0.080099 | 0.079547 | 0.078574 | 0.081169 | 0.077972 | 0.085380 | 9 |
| E-B | 0.104458 | 0.079210 | 0.073629 | 0.071070 | 0.073261 | 0.076062 | 0.065507 | 0.062445 | 0.062358 | 0.077237 | 10 |

Figure 3.1.5 Sample MI Score Hyper-Parameter Output

5. Rank based: Gradient Boosting feature importance ranking

Gradient Boosting algorithm builds up decision trees iteratively and takes into account the loss in the previous iteration. The more a feature is used to make key decisions with decision trees, the higher its relative importance. The importance is calculated for a single decision tree by the amount that each feature split point improves the performance of response variable, which is the total cost in our model. Intuitively, the more times one features are used to split the node in decision trees, the more important it is. The following is a visualization for the number of splits some features are used for, which is also the importance score.

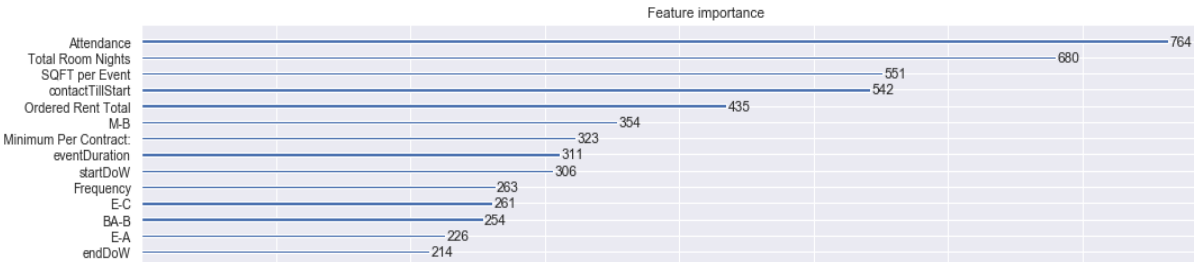



Figure 3.1.6 Part of Feature Importance Output

(vii). Other Methods: Principal Component Analysis (PCA), LASSO and ANOVA

PCA is used for dimensionality reduction which outputs a linear combination of weighted features. The outputs are called principal components (PC) and PC shows how much information (shown by the size of variance) one feature can explain and hence indicates the importance.

Both LASSO and ANOVA are regression based feature selection tools. Basically if there is a strong relationship between the feature we are interested in and the response variable, we conclude the feature is essential. Based on whether $\beta^1$ form the regression model $Y_1 = \beta^1 \cdot X_1 + c$ is 0 or not we can decide if the feature is significant or not for LASSO. ANONA gives F score which indicates the

significance level of each feature by evaluating the linear regression model for each feature, which is similar to the method (iv).

These 3 methods did not produce additional useful information about the feature importance for our data, so the related outputs are not presented here.

## Section II Cost Prediction Model

### Implemented Regression Models

We implemented 3 different regression models to determine the overall best performing model by evaluating $R^2$and mean absolute error metrics. Gradient Boosting was selected as our final regression model after we observed that simple linear regression and elastic net regression did not perform well on the testing sets.

*Ordinary Simple Linear Regression*

We began with Simple Linear Regression and added all second order terms including all combinations of interaction terms from our original feature space, removed insignificant variables one at a time, and analyzed residual plots. We removed the most insignificant feature at each step by observing the corresponding p-values that test $H_0: B_j = 0$, where the highest p-value would be removed at each step. We did not remove a primary feature if it significant in the second order or interaction term even if the primary feature itself is insignificant. Furthermore, we applied Box-Cox transformation on the response variable to induce and satisfy the normality assumption for ordinary least squares. The result of the linear regression model with corresponding residual plots at this point is displayed below.

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | y | R-squared: | 0.695 |
| Model: | OLS | Adj. R-squared: | 0.682 |
| Method: | Least Squares | F-statistic: | 54.19 |
| Date: | Thu, 26 Sep 2019 | Prob (F-statistic): | 5.25e-93 |
| Time: | 13:02:06 | Log-Likelihood: | -1065.7 |
| No. Observations: | 423 | AIC: | 2167. |
| Df Residuals: | 405 | BIC: | 2240. |
| Df Model: | 17 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 9.5222 | 0.295 | 32.270 | 0.000 | 8.942 | 10.102 |
| sqftPerEvent | 1.49e-07 | 1.04e-07 | 1.428 | 0.154 | -5.62e-08 | 3.54e-07 |
| orderedRentTotal | 2.147e-05 | 4.22e-06 | 5.084 | 0.000 | 1.32e-05 | 2.98e-05 |
| Attendance | 3.068e-05 | 1.71e-05 | 1.794 | 0.074 | -2.94e-06 | 6.43e-05 |
| totalRoomNights | -6.364e-05 | 3.59e-05 | -1.772 | 0.077 | -0.000 | 6.98e-06 |
| contactTillStart | 0.0087 | 0.001 | 8.425 | 0.000 | 0.007 | 0.011 |
| M-B | 0.0879 | 0.025 | 3.501 | 0.001 | 0.039 | 0.137 |
| E-A | 0.3525 | 0.182 | 1.933 | 0.054 | -0.006 | 0.711 |
| E-B | 0.2316 | 0.149 | 1.551 | 0.122 | -0.062 | 0.525 |
| E-C | 0.7419 | 0.231 | 3.206 | 0.001 | 0.287 | 1.197 |
| M-A | 0.1002 | 0.025 | 3.984 | 0.000 | 0.051 | 0.150 |
| M-C | 0.0676 | 0.032 | 2.126 | 0.034 | 0.005 | 0.130 |
| FB | 2.951e-06 | 1.37e-06 | 2.149 | 0.032 | 2.51e-07 | 5.65e-06 |
| Attendance^2 | 6.881e-10 | 2.45e-10 | 2.805 | 0.005 | 2.06e-10 | 1.17e-09 |
| orderedRentTotal_Attendance | -2.733e-10 | 8.27e-11 | -3.304 | 0.001 | -4.36e-10 | -1.11e-10 |
| orderedRentTotal_M-B | -3.028e-07 | 7.51e-08 | -4.033 | 0.000 | -4.5e-07 | -1.55e-07 |
| contactTillStart^2 | -3.825e-06 | 4.99e-07 | -7.660 | 0.000 | -4.81e-06 | -2.84e-06 |
| E-C^2 | -0.0653 | 0.022 | -2.923 | 0.004 | -0.109 | -0.021 |

| | | | |
|---|---|---|---|
| Omnibus: | 34.121 | Durbin-Watson: | 1.669 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 48.737 |
| Skew: | -0.588 | Prob(JB): | 2.61e-11 |
| Kurtosis: | 4.176 | Cond. No. | 7.80e+10 |

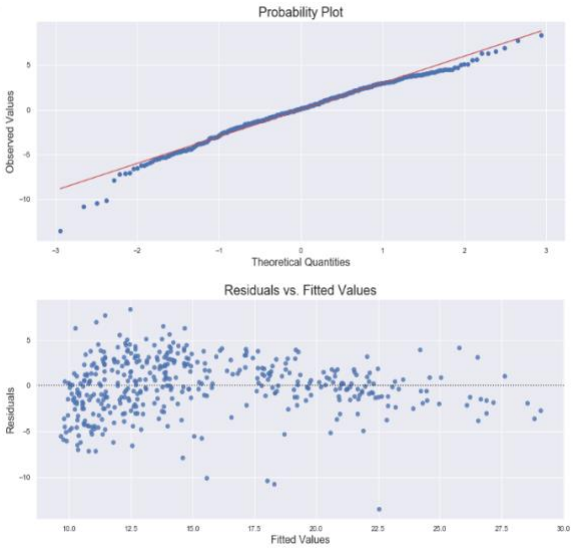
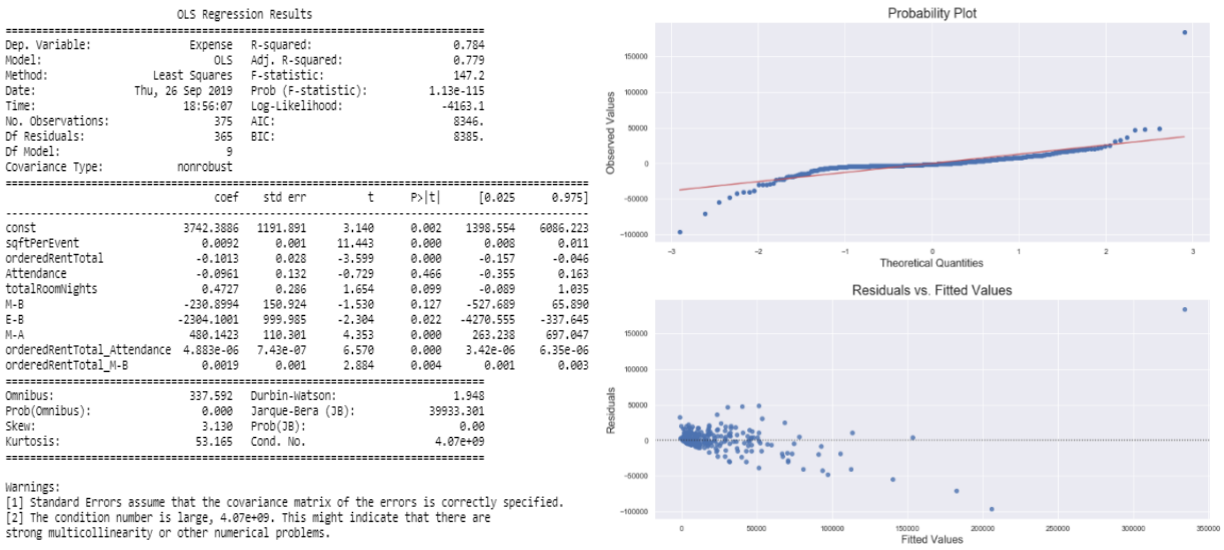



Figure 3.2.1 OLS Regression Model Output and Residual Plots

The Anderson-Darling test statistic corresponds to 1.40 with a critical value of 0.78. Therefore, because the test statistic is greater than the critical value, we reject $H_0$: *Error terms are normally distributed*, and thus the normality distribution assumption for ordinary least squares is violated. Similarly, the residuals vs. fitted values also demonstrate a quadratic-like pattern, so the constant variance of error terms assumption is also violated. We then attempted to remove outliers and removed any remaining insignificant terms, and the regression outputs are below.



OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Expense | R-squared: | 0.784 |
| Model: | OLS | Adj. R-squared: | 0.779 |
| Method: | Least Squares | F-statistic: | 147.2 |
| Date: | Thu, 26 Sep 2019 | Prob (F-statistic): | 1.13e-115 |
| Time: | 18:56:07 | Log-Likelihood: | -4163.1 |
| No. Observations: | 375 | AIC: | 8346. |
| Df Residuals: | 365 | BIC: | 8385. |
| Df Model: | 9 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 3742.3886 | 1191.891 | 3.140 | 0.002 | 1398.554 | 6086.223 |
| sqftPerEvent | 0.0092 | 0.001 | 11.443 | 0.000 | 0.008 | 0.011 |
| orderedRentTotal | -0.1013 | 0.028 | -3.599 | 0.000 | -0.157 | -0.046 |
| Attendance | -0.0961 | 0.132 | -0.729 | 0.466 | -0.355 | 0.163 |
| totalRoomNights | 0.4727 | 0.286 | 1.654 | 0.099 | -0.089 | 1.035 |
| M-B | -230.8994 | 150.924 | -1.530 | 0.127 | -527.689 | 65.890 |
| E-B | -2304.1001 | 999.985 | -2.304 | 0.022 | -4270.555 | -337.645 |
| M-A | 480.1423 | 110.301 | 4.353 | 0.000 | 263.238 | 697.047 |
| orderedRentTotal_Attendance | 4.883e-06 | 7.43e-07 | 6.570 | 0.000 | 3.42e-06 | 6.35e-06 |
| orderedRentTotal_M-B | 0.0019 | 0.001 | 2.884 | 0.004 | 0.001 | 0.003 |

| | | | |
|---|---|---|---|
| Omnibus: | 337.592 | Durbin-Watson: | 1.948 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 39933.301 |
| Skew: | 3.130 | Prob(JB): | 0.00 |
| Kurtosis: | 53.165 | Cond. No. | 4.07e+09 |

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.07e+09. This might indicate that there are strong multicollinearity or other numerical problems.



Figure 3.2.2 OLS Regression Results After Removing Outliers and Insignificant Terms

Similarly here, the normality and constant variance assumptions are clearly violated. Similarly, the $R^2$value seems to be quite high, but when computed against the testing data, the $R^2$ was very poor at 0.2, suggesting signs of significant overfitting and high variance in the model. We then determined that ordinary least squares is most likely not the most appropriate approach for expense modeling.

*Elastic Net Regression*

Because there are significant signs of overfitting in ordinary linear regression, we decided to implement Elastic Net regression using Sci-Kit Learn in order to predict event expenses. Elastic Net regression combines both features of Ridge and Lasso regression, where the loss function is described as below.

$$L_{enet} = \frac{\sum_{i=1}^{n}(y_i - x_i'\hat{\beta})^2}{2n} + \lambda\left(\frac{1-\alpha}{2}\sum_{j=1}^{m}\hat{\beta_j}^2 + \alpha\sum_{j=1}^{m}|\hat{\beta_j}|\right)$$

Equation 3.3

The loss function penalizes the size of the $\beta$parameters to control overfitting and the model complexity. Lasso regression results in some parameters to converge to 0, meaning it contains built in feature selection, whereas ridge regression, using L2 regularization, penalizes the size of these parameters but not necessarily force them to 0. The parameter $\alpha$represents the mixing parameter between ridge and lasso regression, where when $\alpha = 0$, the loss function represents ridge regression and vice versa. The primary parameter to tune for this loss function is the mixing parameter and $\lambda$through cross validation.

We performed a randomized grid search over a set of these parameters by conducting K-Fold Cross Validation. Essentially, the Grid Search algorithm can be summarized in the following image[21], and the algorithm is also explained below:

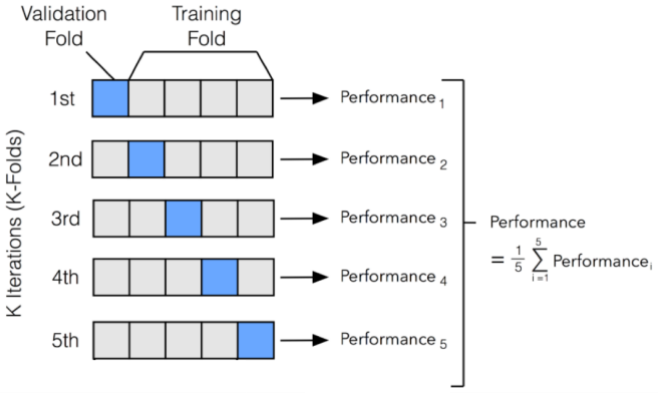



Figure 3.2.3 Grid Search Algorithm

*Randomized Grid Search with K-Fold Cross Validation*

1. Initialize grid of hyperparameters to test over, $H$, and the amount of folds to split into, $K$.
2. For n in range(num_iterations):
    a. Let $h \in H$ represent a random set of hyperparameters to train and validate.
    b. Let $d_h$represent the average performance for hyperparameter set, $h$. Initialize to none.
    c. For $k$ in range($K$):
        i. Split training data into $k$folds.
        ii. Train on $k - 1$folds with $h$ and set $k^{th}$fold as validation fold.
        iii. Assess performance (mean squared error or $R^2$for example) on validation fold.
        iv. Update $d_h$, the average performance, with current iteration's performance value.
3. Choose $h \in H$ from $argmax_h\ d_h$, the best set of hyperparameters with highest validation accuracy.

21 http://ethen8181.github.io/machine-learning/model_selection/img/kfolds.png

Once we determined the optimal set of hyperparameters, we plotted a learning curve that provides details of the performance metric, $R^2$, against the number of samples trained. This gives insight into whether or not the model is overfitting and the comparison between training and testing accuracies.

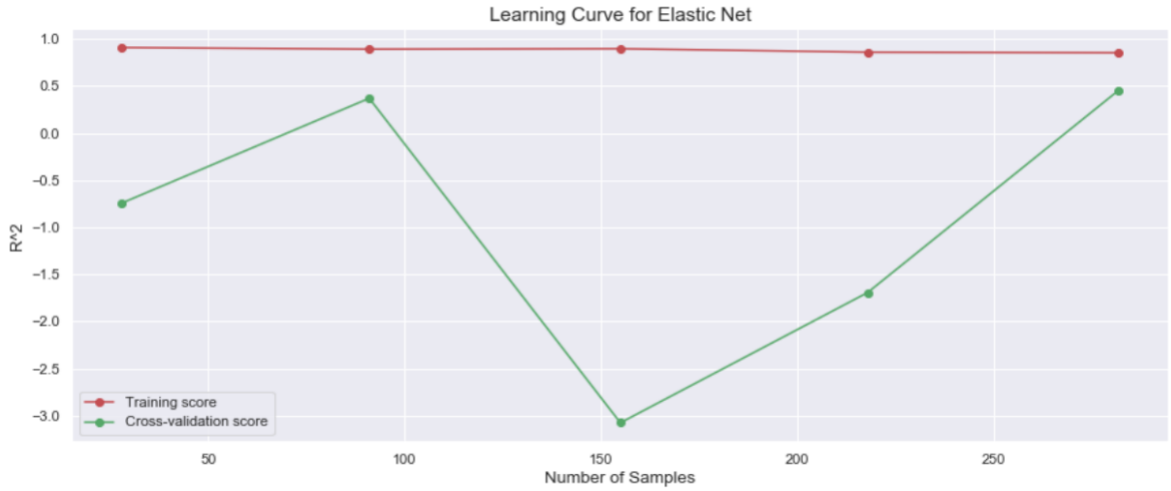
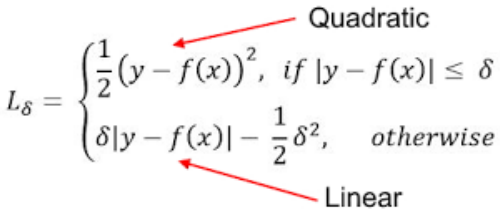



Figure 3.2.4 Elastic Net Regression Training and Testing Curves

Based on the figure above, the Elastic Net regression is highly unstable with small number of samples. We also see that the validation score is significantly lower than the training score. Although this would typically be a sign of overfitting, we can not make this conclusion definitely, as there isn't a consistent pattern where the validation score is significantly lower throughout the entire domain of "number of samples." Essentially, the model may be performing poorly in the validation set due to the limited amount of data, and this model cannot be utilized as our proposed model. The training $R^2$score is quite high, higher than simple linear regression, at 0.85, but this value can not be trusted due to the poor performance in the validation set.

*Gradient Boosting*

Finally, we turned to Machine Learning as a means to attempt to predict expenses more accurately and with less variance than Elastic Net and simple Linear Regression. We used Microsoft's open source LightGBM framework in Python to implement a Gradient Boosting regressor. Gradient Boosting begins with a loss function that needs to be optimized, for example mean squared error. In our implementation, we utilized several loss functions: mean squared error (L2), huber loss, and mean absolute error (L1). Huber loss in particular is used for robust regression models that is less sensitive to outliers than the mean squared error. The huber loss function is presented below.




Equation 3.4 Huber Loss Function

The loss function is penalized quadratically if the error is small and linearly if the absolute error is larger than $\delta$. Therefore, outliers are penalized less (linearly) than mean squared error, which penalizes all errors quadratically. We included the huber loss function in our model because there are some clear outliers where expenses were very large for some feature values that may have induced some overfitting in our ordinary linear regression model.

Once a loss function is determined for the Gradient Boosting Regressor, weak learners (decision trees) are added in succession in order to create a strong final learner by correcting the previous trees' residuals. The splits in the trees are determined in a greedy manner, meaning that the best split is not determined globally at every split, but the best split is determined by metrics such as information gain or scores such as Gini. However, it is critical to ensure that each learner added is a *weak* learner. Adding strong learners together in succession that builds on the previous tree's error can cause a model with very high accuracy (low bias), but high variance and cause significant overfitting. Furthermore, adding strong learners successively is highly computationally expensive, whereas combining weak learners, such as decision trees, require significantly less computational time to train and learn. Functional gradient descent is used to minimize the loss of the objective function when adding successive trees. Each tree is added in succession by first parameterizing the weak learner and setting the parameters of the tree that reduces the residual loss.

In order to ensure that the learner does not overfit, we tuned hyperparameters to optimize the model's predictive power. The following hyperparameters are tuned through the process of randomized grid search as mentioned earlier, and the set of hyperparameters and its purpose are listed below:

1. max_depth : limit the depth of the tree model (avoids overfitting)
2. reg_alpha : L1 regularization
3. reg_lambda: L2 regularization
4. num_leaves : number of leaves in each tree (avoids overfitting)
5. min_child_weight : minimum sum hessian in a leaf or number of instances required to split on a node (avoids overfitting)
6. learning_rate : shrinkage weight

7. subsample : fraction of observations selected for each tree
8. boosting_method : method for boosting procedure
9. objective : L1, L2, and huber loss

Although currently we perform a grid search, this can be quite computationally expensive since it searches over most of the combinations of the hyperparameter grid. Furthermore, we are doing a *randomized* grid search, so we do not find the most optimum set of hyperparameters since it only searches up to n_iter combinations instead of all possible combinations of hyperparameters in the grid. Following the interim, we will utilize Bayesian Optimization in order to choose the most optimum set of hyperparameters.

Grid search is an *uninformed* method: it does not use the results from a previous search to select the next set of hyperparameters. Bayesian Optimization instead uses the previous results to move to a new set of hyperparameters. The optimization problem essentially builds a probability model of the objective function that maps the hyperparameter inputs to a loss. It then selects the next set of hyperparameters by applying a criteria, such as "Expected Improvement[22]" to determine how to move in the hyperparameter space. Essentially, it minimizes the amount of hyperparameter sets to travel over by spending more time analyzing which next set of hyperparameters is best to explore.

Once we determine the optimum hyperparameter set through grid search, we optimized one final hyperparameter: num_estimators, the number of weak learners in the gradient boosting model. Instead of passing this through grid search, which could add to the computation expense, a popular method for optimizing this parameter is conducting a process of "early stopping." This allows us to find the *minimum* number of iterations of weak learners to add that is sufficient to create a model that performs well against out of sample data. The more iterations we conduct, the longer the model takes to train, so determining the minimum number of iterations is highly beneficial in reducing computation runtime.

We first split the training set into $k = 3$folds and begin training on $k - 1$folds. After each individual tree is built, we compute the validation score, and we continue adding trees individually until the validation score does not improve for 50 rounds. We repeat this process until each fold is a validation fold and determine the minimum number of iterations needed, which will be the value for the final hyperparameter, num_estimators. Therefore, the final model for the Gradient Boosting Regressor is trained.

22 https://www.cse.wustl.edu/~garnett/cse515t/spring_2015/files/lecture_notes/12.pdf

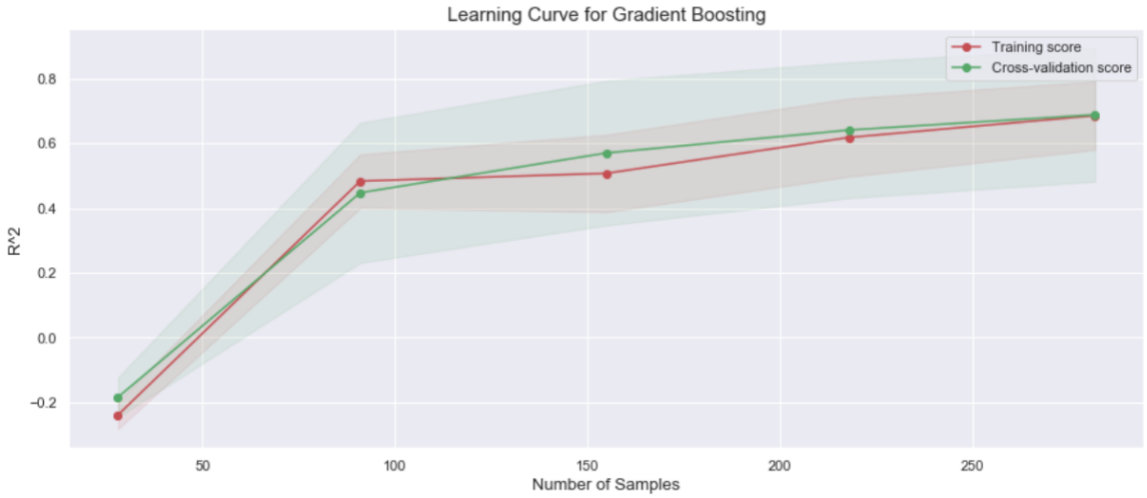



Figure 3.2.5 Gradient Boosting Regression Training and Testing Curves

As observed from the above learning curve visualization, we can observe that the testing and training score seem to converge around each other, which implies there is no significant sign of overfitting or poor model performance. The $R^2$ value hovers around 0.7 for both the training and testing set, and the model similarly performs better in the testing set compared to GWCC's current tool as explained by the deviation percentages. Although there is a somewhat large standard deviation for the testing score, we will attempt to improve this by selecting better hyperparameters and adjust for inflation, and the model will also improve over time as more events are inputted into the model.

| | Predicted | True | % Dev Model | $ Dev Model | Event ID | Budget | % Dev GWCC | $ Dev GWCC |
|---|---|---|---|---|---|---|---|---|
| **4** | 199735.516098 | 199718.60 | 0.008470 | 16.916098 | 14480 | 150214.0 | 24.787176 | 49504.60 |
| **8** | 321378.929048 | 321315.23 | 0.019824 | 63.699048 | 10069 | 108754.0 | 66.153487 | 212561.23 |
| **6** | 97880.148898 | 97508.73 | 0.380908 | 371.418898 | 5501 | 56238.0 | 42.325164 | 41270.73 |
| **2** | 96485.328597 | 96101.75 | 0.399138 | 383.578597 | 7778 | 29495.0 | 69.308571 | 66606.75 |
| **12** | 72908.844896 | 72343.40 | 0.781612 | 565.444896 | 5237 | 62349.0 | 13.815220 | 9994.40 |
| **3** | 119551.569996 | 124592.69 | 4.046080 | 5041.120004 | 6428 | 38610.0 | 69.011023 | 85982.69 |
| **17** | 76510.549529 | 79768.79 | 4.084606 | 3258.240471 | 12009 | 29501.0 | 63.016864 | 50267.79 |
| **15** | 70524.821338 | 73605.27 | 4.185093 | 3080.448662 | 15047 | 39607.0 | 46.189994 | 33998.27 |
| **7** | 158551.995387 | 166842.37 | 4.968986 | 8290.374613 | 6121 | 89671.0 | 46.254060 | 77171.37 |
| **16** | 94742.247675 | 101868.53 | 6.995568 | 7126.282325 | 10833 | 45677.0 | 55.160833 | 56191.53 |
| **0** | 341639.952187 | 295855.02 | 15.475462 | 45784.932187 | 13695 | 191413.0 | 35.301757 | 104442.02 |
| **9** | 283739.484874 | 337039.23 | 15.814107 | 53299.745126 | 6122 | 157126.0 | 53.380501 | 179913.23 |

Figure 3.2.6 Sample Cost Prediction Results Comparison Output

The chart above shows a sample of the events in the testing set. The predicted expense from our regression model ("Predicted"), the true expense ("True"), and the budget expense ("Budget) that is

calculated from GWCC's current tool are displayed above. The deviations are computed above for both our model and GWCC's tool.

*Sensitivity Analysis*

Although the Gradient Boosting model performs well compared to other linear models and is able to predict costs more accurately, it is difficult to interpret the model directly. Therefore, we created Partial Dependence Plots (PDP) that visualize how the cost changes when one feature changes in value and all other feature values stay constant. Note that the PDP outputs one line for each training data, and the yellow highlighted line represents the average from these training data. Furthermore, we used PDP interaction plots to visualize how the interaction of two features can affect cost. We kept our analysis up to the interaction of two features because interactions of three features and higher are either difficult or impossible to visualize.

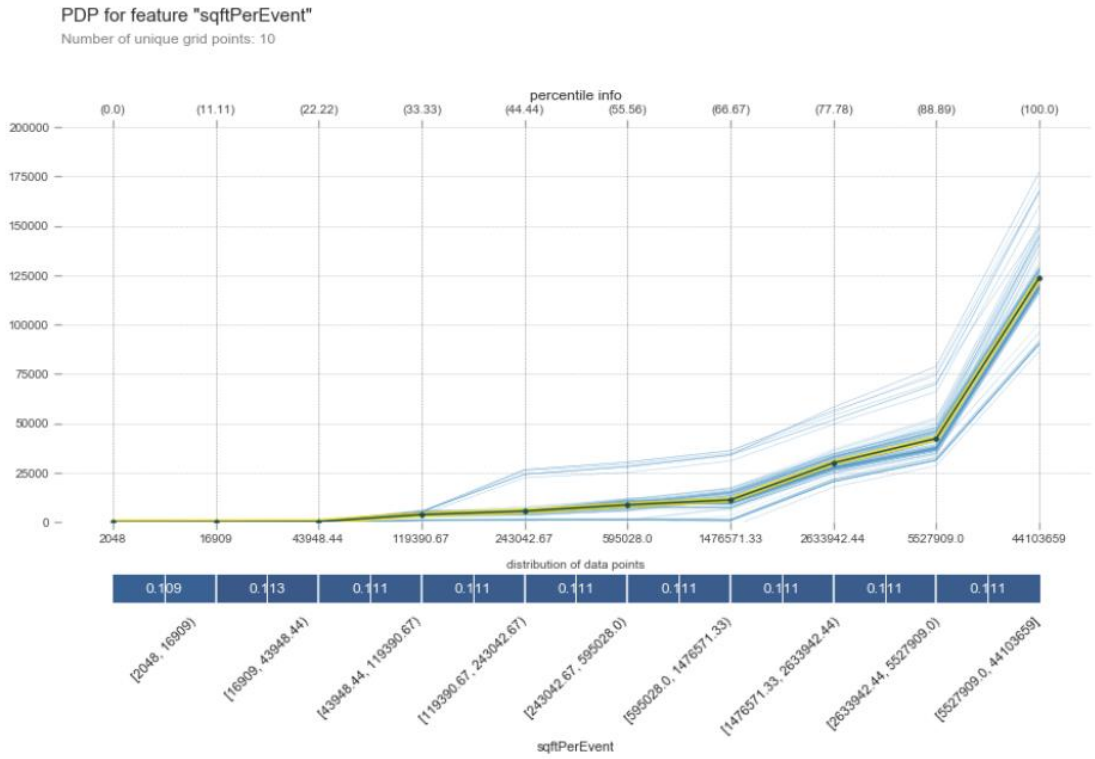



Figure 3.2.7 Sensitivity Analysis for SQFT per Event

The PDP shown above for square foot per event shows us that the cost of an event increases proportionally to the square footage of an event. However, the cost begins to increase at a higher rate once square footage exceeds 1,500,000 and even higher at 5,000,000 square feet. The distribution below the PDP displays that the distribution of square footage per event across the training data is fairly uniform.

The boxplot below is similar to the PDP above but instead of showing one line for each training data, it is instead converted to a boxplot at each interval range of square footage per event.

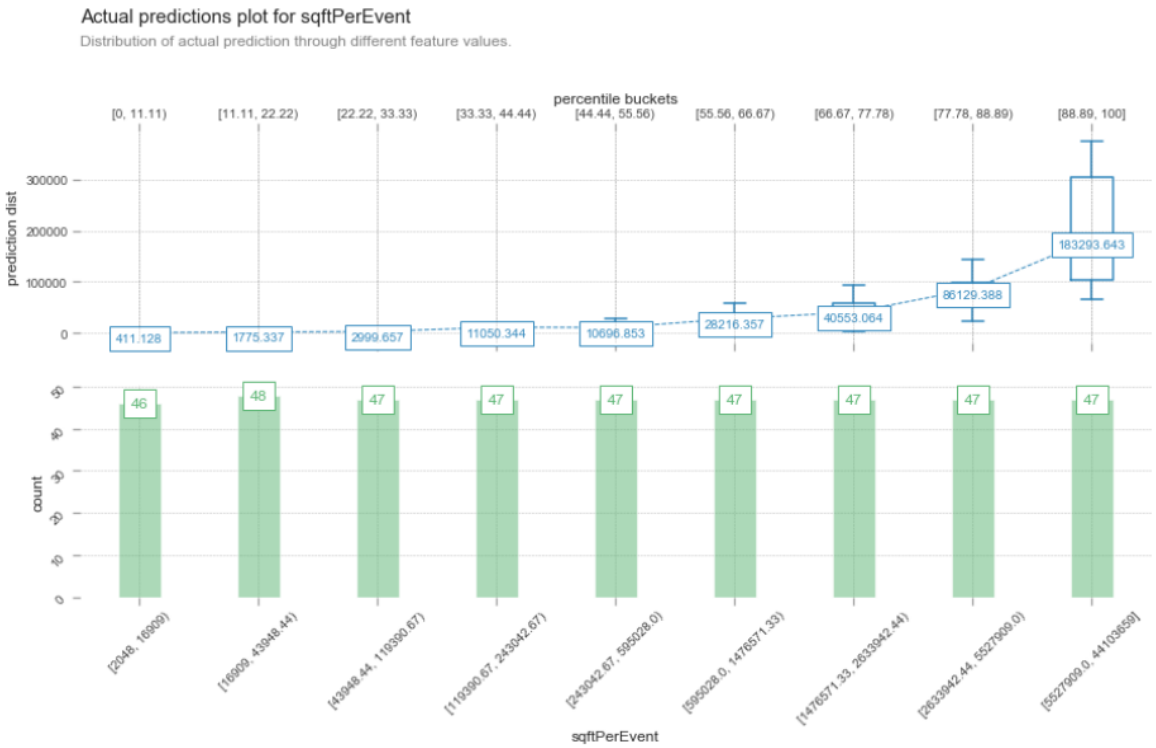
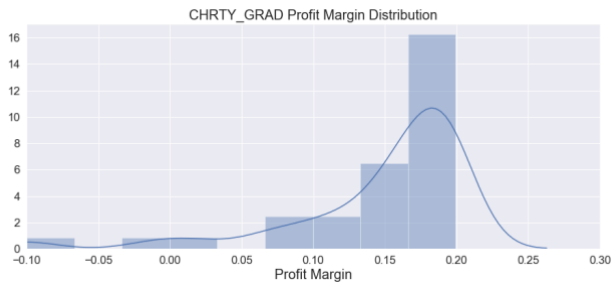



Figure 3.2.8 Sensitivity Analysis for SQFT per Event

## APPENDIX D: Classification

Profit of an event was determined by subtracting the expenses from the revenue made for each event. After mapping the expense and revenue data with the event list and excluding the events occurred outside of GWCC facility, 410 events were used for this model. We generated the adjusted profit margins for the groups by taking the 50th percentile of the distributions and multiplied them to the revenue to get adjusted profits for each event. Comparing the actual profit and the adjusted profit, increase of $2,089.33 per event and total increase of $856,623.35 were identified. Below are the distributions of the adjusted profit margins for each group.



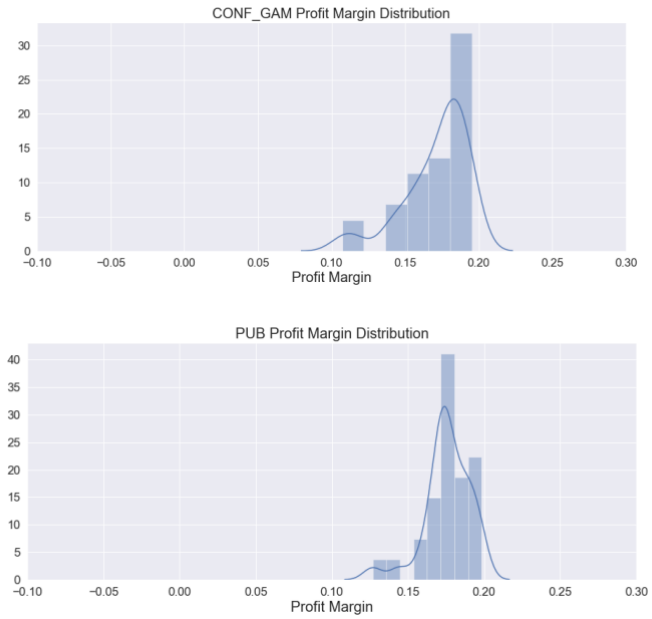



Figure 4.1.1 Sample Profit Margin Distribution Results

| Cluters | # of Events |
|---|---|
| SMALL | 40 |
| CHRTY_GRAD | 37 |
| PUB | 30 |
| CONF_GAM | 30 |
| MEET | 51 |
| ME_OTH | 34 |
| CON_P | 22 |
| FILM_AWC | 31 |
| COMP | 50 |
| CONV | 85 |
| Grand Total | 410 |

| CHRITY_GRAD | Level of Demand |
|---|---|
| 1 | Mid |
| 2 | Low |
| 3 | Low |
| 4 | Mid |
| 5 | High |
| 6 | Mid |
| 7 | Mid |
| 8 | Low |
| 9 | High |
| 10 | High |
| 11 | Mid |
| 12 | Mid |

Figure 4.1.2 Cluster Demand Analysis

## APPENDIX E: Deliverable

| | **Excel** | **GUI** | **Web App** |
|---|---|---|---|
| **Installation Cost** | $0 | $10,000/year* | $10,000/year* |
| **Interactivity** | Challenging | Clear | User-friendly |

| | | | |
|---|---|---|---|
| **Solving Speed** | Slow | Good | Good |
| **Accessibility** | Good | Challenging | Great |

***Cost of Gurobi license for optimization

Table 5.1.1. Tools Comparison

Required event information includes: event type, start date, end date, expected attendance, square foot requested, total room nights, RFP enter date, order rent total, minimum food & beverage revenue, number of exhibit halls, number of meeting rooms, number of auditoriums, number of ballrooms.

## APPENDIX F: Value Calculation

| Month | Profit Sum | Number Of Event | Frequency |
|---|---|---|---|
| 1 | 8867268.77 | 34 | 0.082926829 |
| 2 | 5243012.13 | 33 | 0.080487805 |
| 3 | 11984871.9 | 58 | 0.141463415 |
| 4 | 5619498.98 | 28 | 0.068292683 |
| 5 | 8867714.6 | 40 | 0.097560976 |
| 6 | 3745831.79 | 41 | 0.1 |
| 7 | 3889411.26 | 30 | 0.073170732 |
| 8 | 3233266.81 | 15 | 0.036585366 |
| 9 | 10433158.9 | 31 | 0.075609756 |
| 10 | 6844618.26 | 38 | 0.092682927 |
| 11 | 3118014.28 | 38 | 0.092682927 |
| 12 | 5465811.26 | 24 | 0.058536585 |

| Test Dataset Value Increase | Percentage June - Oct |
|---|---|
| 400000 | 0.37804878 |

| Estimated Annual Increase (K2/J2) |
|---|
| 1058064.516 |

Figure 4.1.2 Extrapolation from Test Data

## APPENDIX G: User Manual

# Table of Contents

**Initial Setup Requirements** **1**

Package Requirements 1

Data Requirements 2

**Web App User Manual** **5**

Log-in 5

Home Page 5

New Event 6

Room Selection 7

Pricing 7

Search Event 8

Calendar 9

**Back-end Maintenance 9**

Data Update 9

# Initial Setup Requirements

## Package Requirements

This help section is broken into *2* sub-categories. The first section details the tools the developers used for all models and general visualizations. The second section details the tools used for the optimization and cost prediction models specifically.

**Note: For both optimization and cost prediction models, we only used Python as our programming language. Thus, the packages provided are only for the use in python. For general requirement, Python, CSS, HTML, Javascript, SQL are used. Only package installations are needed for Python and web-app building.**

### 1. General Requirement

#### *1.* Python 3.6/3.7

| Package | Install Command Line | Usage |
|---|---|---|
| **pandas** | <pip install pandas> | Build up datagram |
| **numpy** | <pip install numpy> | Matrix and array calculation, data process |
| **matplotlib** | <pip install matplotlib> | Visualization tool |
| **scipy** | <pip install scipy> | Statistics and Optimization framework |
| **datetime** | <pip install datetime> | Construct datetime object |

#### *2.* Web-app

| Package | Install Command Line | Usage |
|---|---|---|
| **flask** | <pip install flask> | Build web-app framework |

### 2. Models

| Package | Install Command Line | Usage |
|---|---|---|
| **seaborn** | <pip install seaborn> | Advanced Visualization Tool |
| **plotly** | <pip install plotly> | Interactive Visualization Tool |
| **sklearn** | <pip install sklearn> | Build/select models, metrics, measurements, and other machine learning tools |
| **statsmodels** | <pip install -U statsmodels> | Distribution model analysis and visualizations |
| **math** | Automatically downloaded with Python | Basic calculations |
| **lightgbm** | <pip install setuptools wheel numpy scipy scikit-learn -U> | Machine Learning tool used for regression/prediction |
| **researchpy** | <pip install researchpy> | Cramer's V correlation calculation for feature selection |
| **xgboost** | <pip install xgboost> | Gradient Boosting regression model |

# Data Requirements

This section describes all necessary *.csv or .excel* files needed ensure the back-end models can run. The room recommendation optimization model requires 2 csv files. The cost prediction regression model requires 1 csv file. The structure each file and the self-defined columns will be explained in this section.

## 1. Optimization Model

#### *1. Room Data.csv*

**Sample:**

| RoomID | Name | SQFT | Cost | Building | Floor | X | Y |
|---|---|---|---|---|---|---|---|
| 0 | Exhibit Hall A1 | 149,000 | 2,859 | 0 | 0 | 1 | 1 |
| 1 | Exhibit Hall A2 | 86,000 | 2,204.77 | 0 | 0 | 2 | 1 |

***RoomID*:** Unique to each room.
***Building*:** 0 represents Building A, 200 represents Building B, and 260 represents Building C. Those are parameters representing the weights.
***Floor*:** First Floor is 0, Second Floor is 30, Third Floor is 60, Fourth Floor is 90, Fifth Floor is 120. Those are parameters representing the weights.
***X & Y*:** Represent the horizontal and vertical location of each room on each floor.

***2. Occupied Rooms Data.csv***

**Sample:**

| RoomID | Name | DateIn | DateOut |
|---|---|---|---|
| 1 | Exhibit Hall A2 | 2014/12/31 | 2015/1/4 |
| 41 | Exhibit Hall B2 | 2014/12/31 | 2015/1/4 |

**Note: This above chart is the overall occupancy status for all the rooms in historical dataset. If one event used multiple rooms, all rooms should be recorded separately. The mapping of RoomID to Room Name is documented in the Room Data.csv file.**

## 1. Cost Prediction Model

***Event Data Cleaned.csv***

**Sample:**

| EventID | SQFT | Ordered Rent Total | Min F&B | Attendance | Total Room Nights | Event Length |
|---|---|---|---|---|---|---|
| 6179 | 1,284,565 | 67,500 | 0 | 19,000 | 10,717 | 3 |
| 5058 | 4,764,175 | 139,825 | 100,000 | 7,114 | 9,732 | 5 |

**Sample continued:**

| Contact Till Start | E-A | E-B | E-C | M-A | M-B | M-C |
|---|---|---|---|---|---|---|
| 574 | 1 | 1 | 0 | 6 | 11 | 0 |
| 734 | 0 | 1 | 0 | 1 | 51 | 0 |

***Attendance*:** The attendance input feature is not the same figure as the actual attendance, for this information will not be available until the event is over. Thus, we used the Expected Attendance provided in RFP as the input for the attendance feature. In historical data, if an event did not have Expected Attendance information, the actual attendance was used instead to keep as many training data as possible.
***Contact Till Start*:** This feature is created by the developers by taking the difference between the start date of the event and the submission date of the RFP.
***E-A, E-B, E-C*:** "E" represents Exhibit Halls, and "A" represents Building A. All tags represents the number of Exhibit Halls used/chosen in the given event.
***M-A, M-B, M-C*:** "M" represents Meeting Rooms, and "A" represents Building A. All tags represents the number of Meeting Rooms used/chosen in the given event.

**Note: We recommend to make Expected Attendance a required information in the future events' RFP so a more accurate estimation can be obtained. The room information in our model is given by the outputs of optimization model as described in the last section.**

## 1. Classification Model

No initial data required, for all clusters and profit margins were already fixed and calculated in the back-end. However, the model can be updated if a more rigorous demand forecasting tool is available. The basic logic behind the model is to assign reasonable and profitable profit margins to each event based on type and months. For different clusters, the demand peak, trough and average seasons are also differentiated, so the demand are analyzed within one cluster and suggested profit margins vary based on the cluster and demand.

# Web App User Manual

## Log-in

1. **Inputs Requirement**

   <EmployeeID> and <Password>; you can choose to view or hide your password input. The inputs are case sensitive.

2. **Add New Users**

   The only current user in the initial system is the one with username <Mark> and password <apple>. In order to add or remove employee IDs, the database *5GWCC.db* needs to be updated. Follow this procedure in order to add an employee:

   *1.* Open the *Command Line*.

   *2.* Open the folder in which the web app is stored. For example, if the web app is in the folder *Documents/Georgia Tech Senior Design*, the command should be: $cd Documents/Georgia Tech Senior Design

   *3.* Then type $sqlite3 5GWCC.db and press enter, at which time the database is loaded.

   *4.* To view all employee IDs and passwords, type SELECT * FROM user; and press enter.

   *5.* To add an employee ID, type INSERT INTO user (employeeID, empPassword) values ("NEW EMPLOYEE ID", "NEW EMPLOYEE PASSWORD"); and press enter.

   *6.* To ensure it was properly added, view the list by typing SELECT * FROM users; and press enter.

   *7.* In order to remove an employee, type DELETE FROM user WHERE employeeID = "DELETED EMPLOYEE ID" and press enter.

   *8.* To ensure it was properly removed, view the list by typing SELECT * FROM users; and press enter.

## Home Page

After logging in, the web-app will redirect to the home page. From the home page, there are three options: to insert a new event into the model in order to output cost, room selections, and profit margin; to look up a previous event; or to view a calendar. Please select the appropriate option and the page will redirect. At any time, you also have the option to log out of the system by clicking "Logout" in the navigation bar. The logic of the navigation is given in the flowchart shown below.

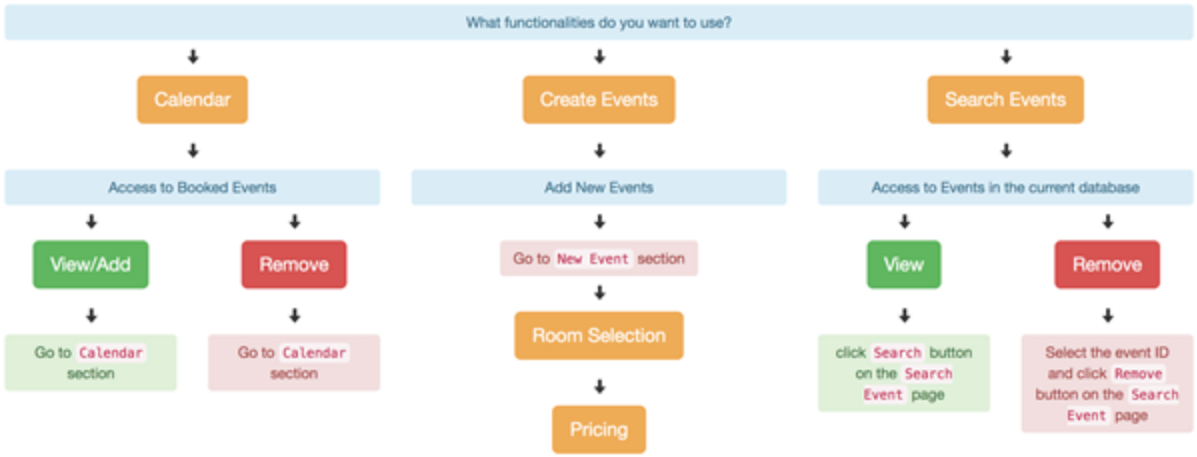



**Fig.1 Home Page Logic Flowchart**

# New Event

1. **Required Inputs**

| | | |
|---|---|---|
| **Req** Event Name | **Req** SQFT Requested | **Req** Number of Exhibit Halls |
| **Req** Event Type | **Req** Total Room Nights | **Req** Number of Meeting Rooms |
| **Req** Event Start Date | **Req** RFP Enter Date | **Req** Number of Auditoriums |
| **Req** Event End Date | **Req** Order Rent Total | **Req** Number of Ballrooms |
| **Req** Exp Attendance | **Req** Min F&B Revenue | **Opt** Min SQFT of all rooms |

2. **Navigate to Room Allocation Page**

If all of the input information is available, pass it into the Event Information form. Press `submit` when finished.

If a *feasible* solution is produced, the web app will redirect to the room allocation solution. If there is an *infeasible* solution, a message line will appear with the option to return to the input page.

# Room Selection

1. **View Results**

   In order to see which rooms are recommended for an event, hover over the picture for the building. The rooms chosen by the optimization model are presented in the table. Only rooms that are unbooked at the time when inputs are entered will be recommended.

2. **Add Rooms**

   To add a room to the existing output table, select a room from the drop down list and then *click* the plus symbol.

   **Note:** The drop down list includes all the rooms in that building. This will allow any room to be selected regardless of if the room is already booked for another event.

3. **Remove Rooms**

   To remove a room from the existing output table, *click* on the red negative sign next to the room's name.

# Pricing

3. **Cost Prediction Result**

The entire model is already trained in back-end, so once we called the model by passing in all the features along with the chosen room results, the cost prediction model is automatically running and the final outputs are shown in the *Cost Prediction & Pricing* page. A range of predicted cost is shown where the lower and upper bounds are given based on 90% of confidence interval. The user has the opportunity to move the sliders to try different combinations of the feature inputs, and the cost prediction and the baseline price results will be automatically changed at the same time.

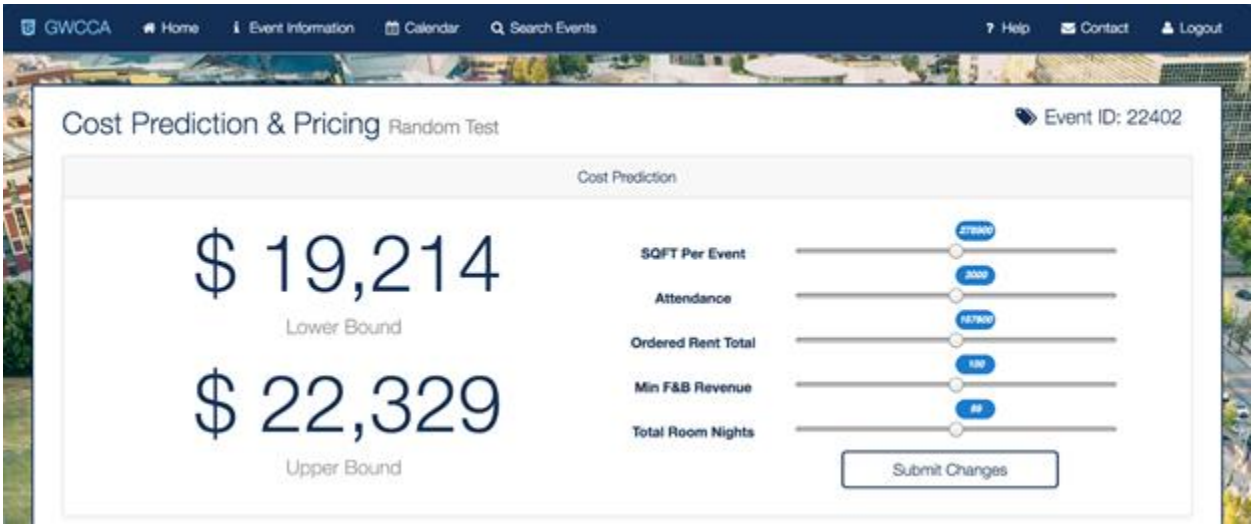



**Fig.2 Cost Prediction Page Demo**

**Note:** As more and more event-based, accurate, and complete data gathered and updated in the database, the cost prediction is guaranteed to give more accurate results.

4. **Dynamic Pricing Model**

On the right side of this section on the pricing page, there is an *interactive profit margin distribution chart*, which allows the users to move the expected or desired profit margin. The lowest profit margin is given as default, which is derived based on demand and type cluster. The user cannot drag the cursor to the left side of the default lower bound, for there is a high risk of not making enough profits.

On the left top of this section, as the user changes the expected/desired profit margin on the interactive distribution chart, the corresponding suggested *Baseline Price* changes at the same time. On the bottom-left of this section, there is a suggested profit margin which is calculated by the classification model. Same type similar season events will be suggested the same profit margin.

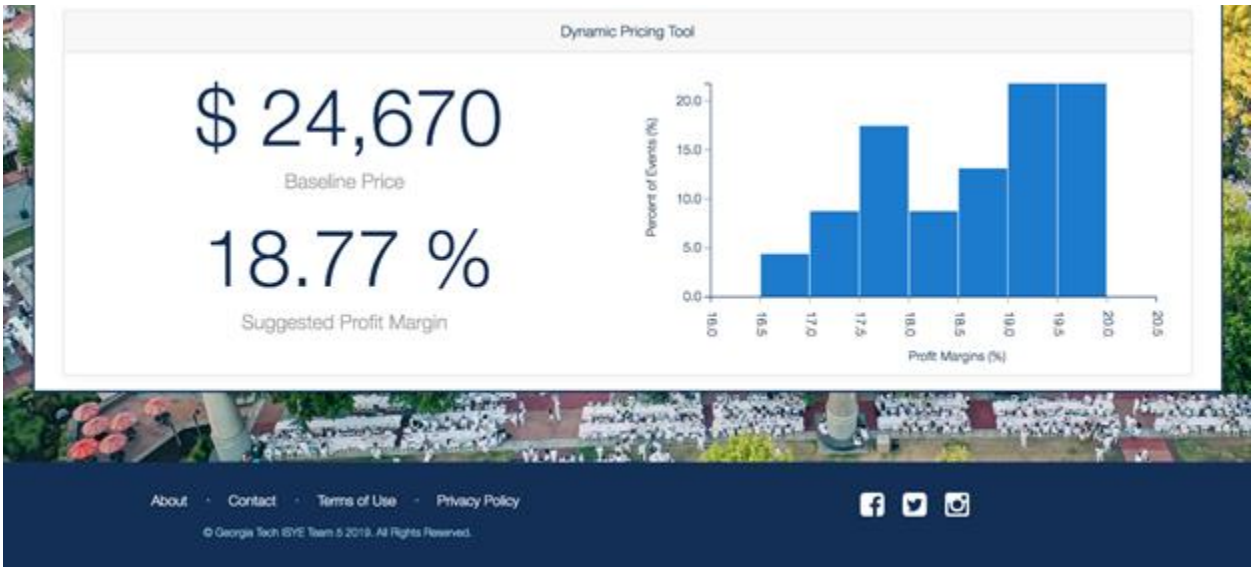



**Fig.3 Pricing Page Demo**

# Search Event

In order to search for an event, pass in the event ID or event name in the search bar and click search. The event information will load. Except the original inputs user entered in the New Event page, the output also includes the cost prediction and baseline pricing results.

The user will be able to remove the events from the database if the event is no longer desired anymore. The user will also be able to confirm the booking, so all the event information will be added to the back-end database. After multiple events were passed into the system, the user is able to export all past events into allEvent.csv

Note: All events that are previously entered into the web app will be saved in the system. All the historical events from Jan 2015 to August 2019 are stored in the database as well. Whenever the user presses Export All Events.csv, All the historical and the newly-added events will be exported. The exported .csv can be used as the training dataset for Cost Prediction. The details for updates can be found in Data Update section.

## Calendar

1. **View Booked Events**

After clicking the Calendar button on the navigation bar or directly navigate from the home page, the calendar page will be presented. The current records are the events that have already been booked, including both historical ones and some of the future ones which are guaranteed to happen.

2. **Add/Remove Events**

The calendar page has a *Google calendar* embedded. In order to add or remove events from the calendar, the user can do it here and the results are automatically synced to the Calendar page. For further help with the Google calendar, click here.

# Back-end Maintenance

## Data Update

This section, we will introduce how the data described in Date Requirement section can be updated. Thus, the model can be re-trained based on the most up-to-date database, which will give as accurate prediction as possible. Once GWCCA owns more accurate cost per room data, it also has the opportunity to replace the estimation for cost per room we obtained.

1. **Optimization Model**

**1. Room Data.csv**

Update in the <Room Data.csv>

1. Open the Room Data.csv
2. Room ID and Name are paired and should NOT be modified.
3. SQFT and cost can be updated as needed/new information comes in.
4. Once all new numbers are entered, make sure to save the changes in the .csv file.

View/Update in the <Database>

1. Open the Command Line.

2. Open the folder in which the web app is stored. For example, if the web app is in the folder Documents/Georgia Tech Senior Design, the command should be: `$cd Documents/Georgia Tech Senior Design`

3. Then type `$sqlite3 5GWCC.db` and press enter, at which time the database is loaded.

4. To view the updates in .csv file, type `.read 5GWCC.sql` and press enter.

5. To view the updated room information, type `SELECT * FROM room;`

**2. Occupied Rooms Data.csv**

Update in the `<Occupied Rooms Data.csv>`

1. Open the Occupied Rooms Data.csv

2. The user can update booked rooms by entering the RoomID, corresponding Room Name, DateIn, and DateOut. The record should be obtained from newly booked events.

3. Once all new numbers are entered, make sure to save the changes in the .csv file.

View/Update in the `<Database>`

1. Open the Command Line.

2. Open the folder in which the web app is stored. For example, if the web app is in the folder Documents/Georgia Tech Senior Design, the command should be: `$cd Documents/Georgia Tech Senior Design`

3. Then type `$sqlite3 5GWCC.db` and press enter, at which time the database is loaded.

4. To view the updated occupied rooms information, type `SELECT * FROM booked;`

**Note:** It is possible that insert errors occur for past occupied rooms data because of duplicated event entries in *Underboeck*. Those errors do not affect the other entries in the table. Please ignore them and continue.

1. **Cost Prediction Model**

**Event Data Cleaned.csv**

Update in the `<Room Data.csv>`

1. Open the Event Data Cleaned.csv

2. Enter all the entries needed in the .csv.

3. Open the Python Jupyter Notebook file `<Cost Prediction Model.ipynb>`.

4. Run the block of code under `Preprocessing Data` and `Gradient Boosting Hyperparameter Tuning`.

5. To make sure the model is updated, run the following block of code, and to see if the results are different from the previous version:

```
print("R^2 is " + "\t" + str(round(r2_score(y_test, y_pred), 4)))

print("MAE is " + "\t" + str(round(mean_absolute_error(y_test, y_pred), 2)))

print("MSE is " + "\t" + str(round(mean_squared_error(y_test, y_pred), 2)))
```

6. Copy the results from running `<grid2.best_params_>`, which should output a dictionary.

7. Run the next block of code. In each of the 3 folds, record the iteration of the best iteration.

8. Take the average of these 3 values and round to the nearest integer.

9. Append the rounded average in the above dictionary with a key of n_estimators.

10. The dictionary will contain the following keys:

| | |
|---|---|
| **Req** subsample | **Req** min_child_weight |
| **Req** reg_lambda | **Req** max_depth |
| **Req** reg_alpha | **Req** leaning_rate |
| **Req** objective | **Req** boosting |
| **Req** num_leaves | **Req** n_estimators |

Update in the <Back-end>

1. Open the 5GWCC.py in an editor.

2. Ensure that in line 209, the updated .csv is being used. Change the first parameter, X_train.csv to the name of the updated .csv file if necessary.

3. In line 212 under the <lgbParams> variable, replace the dictionary with the results from <grid2.best_params_ >and <n_estimators>.

## References

"Compelling Guest Experiences" (2019). Georgia World Congress Center Authority. https://www.gwcca.org/