Technical Memo

Nemacolin Woodlands Resort
X
CMU MHCI Woodlanders

July 30, 2019

Table of Contents

Introduction

Nemacolin offers a variety of lodging and recreational options to its guests. The team's research has shown that this can often cause confusion for prospective guests and prevent them from booking.

Since January 2019, the team has been working on designing a solution that would curate offerings for the guests and present the information they need to easily make a reservation online. In order to design this solution, we used a mix of qualitative and quantitative research.

This document summarizes the team's quantitative research and presents the findings. It also outlines how Nemacolin could leverage Machine Learning on the data it collects about the guests in the future. The reader of this document is assumed to be familiar with the different software systems used by Nemacolin for its day-to-day operations.

We used the reservations data present in HOST and the call center data available on NAVIS to identify guest patterns. **Section I** discusses the various datasets used during the project to gain a better understanding of how Nemacolin operates and identify opportunities for design solutions.

Since there was a lot of data about the guests, the team wanted to explore the possibility of making recommendations to guests automatically through Machine Learning. **Section II** provides a primer on recommender systems. It also details how we trained a model to classify guests into five different hotel types and the limitations of the datasets that Nemacolin currently has.

Due to the limitations of the data, we were not able to prototype a fully autonomous recommender. We instead relied on our domain knowledge to develop the logic that our prototype uses to make recommendations. **Section III** talks about this process and the logic behind our prototype.

Finally, **Section IV** outlines how this prototype could be converted into a production-ready system. Before the prototype can be made production-ready, there are a few technical challenges that Nemacolin must address. This section also presents some suggestions on data collection strategies for the future.

Section I
Analyses of the datasets

Over the course of the project, the team received three primary datasets about the guests from Nemacolin:

- **Reservation Data:** This data comes from the HOST property management system. It contains information about all of the room reservations made at the resort. A part of this dataset also includes psychographic profiles on the guest. These profiles were created by Acxiom.
- **Guest Feedback Data:** This is the data on the Medallia platform which is used extensively by the resort to ensure the quality of service provided to the guest. The platform collects responses to post-stay surveys. It also provides access to guest reviews about the resort.
- **Call Center Data:** Nemacolin uses the NAVIS platform for managing calls received from guests. This platform has a lot of data related to the calls and reservations made. There is some overlap between the data found on this platform and the reservation data. However, there are certain pieces of information about the guests on NAVIS that are not found in HOST.

Additionally, the team received 2VS reports from InfoGenesis, the platform used to manage finances at the resort. These reports contained information about every transaction made at the resort from 2016-2018. However, it did not contain an itemized list of goods and services for each of these transactions. As a result, the team was unable to use this dataset in the project.

The three datasets served as a starting point for the team's quantitative research and helped determine the target guest segment for the project. Analyzing the datasets also provided insight into what kinds of recommender systems could be built. The final prototype works around the limitations of the data mentioned in **Section II**. And, **Section IV** provides recommendations on how these limitations can be addressed to build more effective Machine Learning systems in the future.

Reservation Data

ABOUT THE DATASET Nemacolin uses HOST as the property management system. The team received HOST data about guest reservations between 2017-2018. This data also contained some psychographic information about the guests. While demographic and psychographic information about the guests was useful in understanding the market, it did not correlate very well with their behaviors. This limited the amount of information we could use in our Machine Learning models.

This dataset was used to build the first version of the recommender prototype. This recommender prototype was a proof of concept to determine the possibility of using Machine Learning in the final solution.

LIMITATIONS AND FUTURE WORK The dataset consisted only of the room reservations made by the guests and did not include any restaurant or activity reservations. As a result, the team could not build a Machine Learning model that could predict guest behavior with respect to restaurants or activities. Machine Learning looks at historical data to identify patterns in guest behavior. The dataset lacked this historical behavior for activities and dining, thus no patterns could be identified for those categories.

Guest Feedback Data

ABOUT THE DATASET Medallia is used by Nemacolin to ensure that the guests receive quality service. Nemacolin sends out post-stay surveys to guests via Medallia, and the platform collects data on guest demographics, satisfaction ratings, and other comments about their stay.

This dataset contributed to identifying the target demographic for this project along with the reservation data. Since this dataset primarily consisted of natural language data, the team utilized it more qualitatively as building a Natural Language Processing system would be a time-intensive endeavor.

LIMITATIONS AND FUTURE WORK Since medallia aggregates a lot of reviews from the guest, it is a good repository for using Natural Language Processing algorithms (a subset of Machine Learning specifically focused on interpreting the human language to make predictions). This was outside the scope of this project and the team used this platform only to understand the domain. However, the data available on this platform could be used in the future to better understand how Nemacolin positions itself in the industry. It could also be used to perform sentiment analysis and automatically process data at scale.

Sentiment Analysis helps process large volumes of unstructured data such as reviews and automate business processes and gain actionable insights. It can help identify situations in which angry customers that are about to write a lot of bad reviews about the business online.

Call Center Data

ABOUT THE DATASET The data present on the NAVIS platform overlaps with the one present in HOST. One of the reasons for this overlap is that the Reservation Specialists have to enter the same information into both NAVIS and HOST when guests call to make reservations. In addition, the Reservation Specialists also have to copy over some information from HOST into NAVIS. This manual process introduces a possibility for human error when the information is copied over.

However, there are certain pieces of information about the reservations such as the reason for visit which are unique to NAVIS. The final prototype uses this dataset to predict which hotels should be recommended to a guest. Even though there is slightly more information about the reservations on this platform, it doesn't contain any data on reservations related to restaurants or activities.

LIMITATIONS AND FUTURE WORK Research into the platform has shown that NAVIS has the potential to be an excellent data collection tool. It already collects a lot of unstructured data in the form of call recordings and notes from Reservation Specialists. By collecting a few more pieces of data at the end of each call can allow Nemacolin to build Machine Learning systems that are capable of personalizing the recommendations that the guests receive for restaurants as well as activities.

Section II

Machine Learning Systems

Recommender Systems

A recommender system is a method for filtering information to present users with only the most relevant items. They are widely used in industry for a variety of applications such as product recommenders on Amazon and movie recommendations on Netflix. Recommenders help provide users of these systems a more personalized experience.

There are two primary types of recommender systems. The first type is called Collaborative Filtering and the other is called Content-Based Filtering. **Figure 1** highlights the difference between the two types of recommender systems.

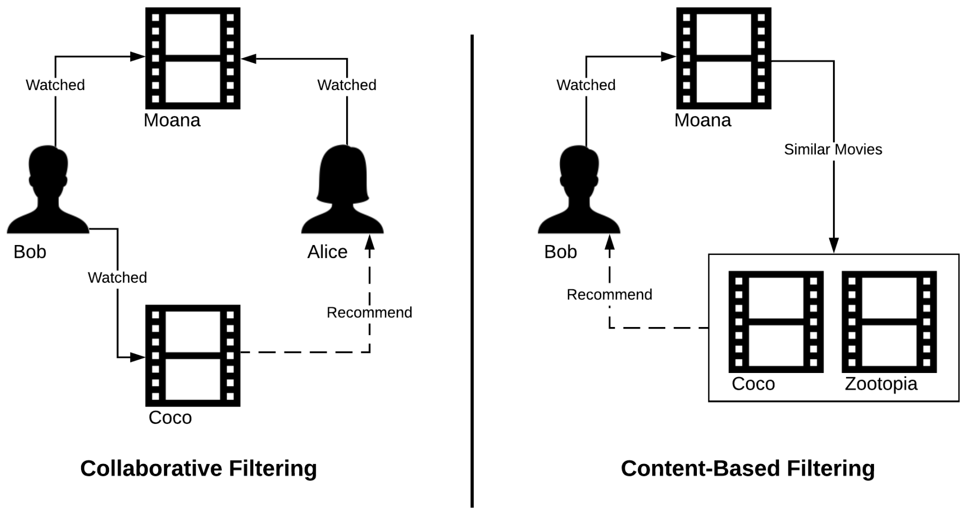


Figure 1: The two types of Recommender Systems

COLLABORATIVE FILTERING This algorithm builds a model of users' past behavior and compares it with the behavior of other users to make recommendations (e.g., Netflix). For example, let's assume the algorithm discovers that Alice and Bob have similar preferences when watching movies. If both the users watched Moana, the algorithm can then recommend movies to Alice that she has not yet seen based on what Bob has seen. In this case, that movie is Coco.

CONTENT-BASED FILTERING This algorithm makes use of the characteristics of items to make recommendations of similar items. For Example, if Bob watches Moana, then the algorithm can recommend movies such as Coco and Zootopia if they have similar characteristics such as genre, production studio etc.

Since our target segment are first-time guests, we do not have access to their past behavior or their preferences. As a result, neither of the two classical recommender systems can be used. We instead employ a Decision Tree to predict a guest preference for a hotel and then make a "recommendation". Thus, the model that we use is not a true recommender system but a classifier. A **Machine Learning Classifier** attempts to map a given input to a category within a predefined set. For instance, we try to map the guest data to the different hotels at the resort.

Decision Tree Classifier

Our concept makes use of the Decision Tree algorithm to classify guests into different hotels. The decision tree is named so because it can be equated to asking a series of questions. The questions asked depends on the responses received previously. This is illustrated in **Figure 2**. Each internal node of the Decision Tree represents a question on an attribute of the guest data. The data is split between the different branches of this node based on the answer to this question. The leaf nodes (the nodes in the tree that do not branch into any other nodes) represent the final category (the hotel) to which a given input belongs.

For example, when someone asks for a recommendation for a two-night couple's getaway, the recommender first checks how many people the reservation is being made for — two, in this example. Based on this response, the next thing it checks for is whether the trip is a family vacation or not. Following the same procedure, the decision tree arrives at a leaf node which has Chateau as its decision. This decision is then used to recommend the Chateau to the guest.

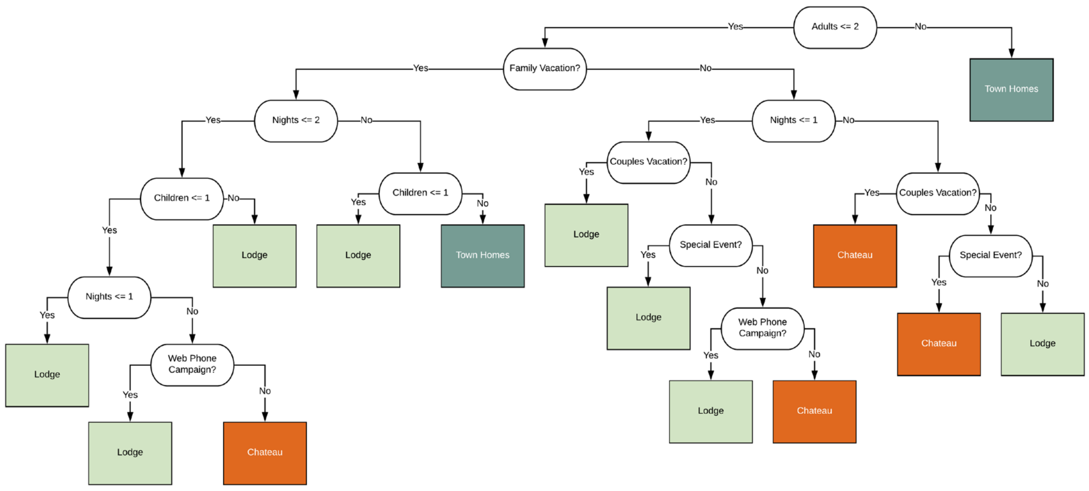


Figure 2: A simplified Decision Tree

In our concept, the Decision Tree is trained using historic guest data in order to predict the hotel the guests stayed at. At first, all of the features in the dataset are considered and the feature that contains the highest amount of information gain is selected as the root. The rest of the features are used to recursively create more internal nodes for the Decision Tree in order to maximize the information gained. Once the model is generated using the historic guest data as the training data, it can be used to predict the hotel for future guests.

AVOIDING OVERFITTING Overfitting is a type of error in which a model becomes overly complex to explain the idiosyncrasies present in the data. Consequently, the model is unable to perform well on unseen data. When using Machine Learning models, it is very important to ensure that the model does not **overfit** the training data, i.e., the model is capable of predicting relatively accurately for unseen data. To prevent overfitting in our model, we tune one of the parameters of the model that allows us to control how deep the decision tree can grow. As a result, the model doesn't have to ask the users too many questions before making a recommendation. This in turn leads to a better experience as well as recommendations that are generalizable to new guests. **Figure 3** shows the change in accuracy for two parts of the dataset when we tune one of these parameters.

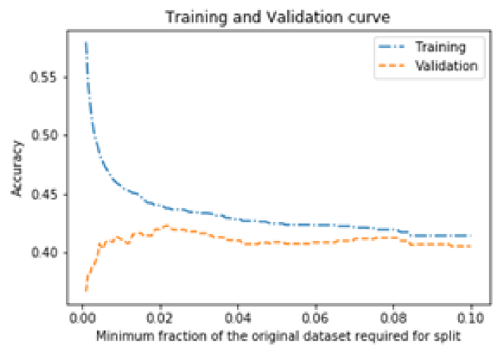


Figure 3: Change in accuracy when tuning hyperparameters

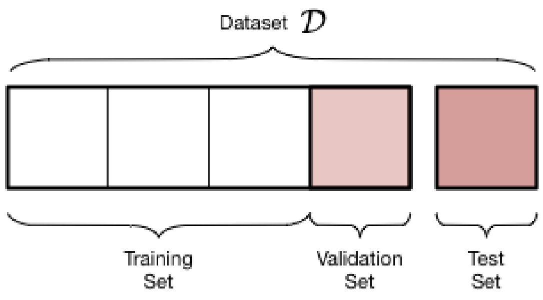


Figure 4: An example of how datasets are split in Machine Learning

When building a Machine Learning model, the original dataset is usually divided into three parts — the training set which is used to create the decision tree model, the validation set which is used to tune the parameters, and the test set which is used to report the results (Shown in **Figure 4**). Since we were determining how deep the decision tree would be, **Figure 3** is a comparison between the training and validation sets. This ensures that the data used to test the accuracy of the model in now way used to train the model.

Limitations of the data

Machine Learning algorithms essentially generate an output based on some input that they receive. In our case, the ML algorithm takes as input different pieces of information — also known as features — about the guest and outputs a recommendation for which hotel the guest should stay at. The ML algorithms find patterns in the data automatically to draw insights, and make decisions or predictions. Having different features can help identify more patterns in the data which can lead to better predictions.

There were a few limitations in our dataset that prevented the model from achieving a higher accuracy. The limitations listed below are the most important:

» **Limited Number of Features** The dataset had a limited number of features that were relevant to making predictions on user behavior. This was one of the major shortcomings of our Machine Learning prototype. Out of the different features available to us, we were able to identify which ones contained the most information about the guests. However, there was no way for the team to identify which important features were missing through quantitative research. The team instead relied on qualitative research to identify the missing features.

» **Data Silos** There was not enough granular information available about the guests. In some cases, when this information was present, it was not accessible enough to be used in building Machine Learning models. One such instance of the data not being accessible was the data about items and services purchased by individual guests at Nemacolin. This data could have helped in building a recommender system based on guest purchase behavior.

» **Skewed Dataset** The dataset was skewed since there were more training examples for certain classes (Lodge & Chateau). As a result, the model was biased towards recommending these classes at the cost of the other classes (Falling Rock).

Limitations of Machine Learning Model

The team decided to use the Decision Tree algorithm in the Machine Learning model. One of the primary reasons for doing so was that the decisions made by the Decision Tree algorithm are easily visualized and readily understandable. However there are some limitations to this algorithm:

» The algorithm is susceptible to skewed datasets and creates trees that are biased towards recommending the dominating classes in the dataset. Since the number of reservations at the Lodge and Chateau were much higher compared to Falling Rock, the decision tree generated by our model was biased towards recommending those hotels.

» It is hard for the algorithm to model certain patterns in the dataset that other algorithms can model easily. This is challenging to explain but generally speaking, other algorithms are capable of discovering more abstract patterns.

» This algorithm requires manual feature engineering (the process of transforming existing features into new ones). Other algorithms such as neural networks do not require this.

The team recommends building out a Machine Learning system which uses a combination of different algorithms (commonly known as ensemble systems). This way, the system is more robust and can usually generalize better to unseen data.

Section III
Logic of the Prototype

Since the datasets could only help with predicting the hotels for the guests, Machine Learning alone did not suffice the requirements of our prototype. The prototype had to provide prospective guests with a complete and simplified booking experience, thus the logic had to be more extensive that a singular decision tree.

The team utilized the insights from the qualitative research and data analysis to augment the generated Decision Tree model. This augmented Decision Tree was used in the final prototype to provide recommendations for activities and dining experiences in addition to hotel rooms. This logic served two objectives:

1. Provide a complete planning experience, including activities and dining
2. Gather the minimum and most influential pieces of information about the guest

Process

As mentioned previously, the team used a mixed-methods approach for building the logic of the final prototype. This consisted of three core steps:

1. Build upon the Decision Tree Classifier

We used the Decision Tree classifier as a starting point for creating our prototype logic. The model we used revealed features that proved impactful in predicting hotel type from our classifier. We incorporated the following features into the Preferences section of our prototype:

» Number of adults
» Number of children
» Month of stay and length of stay (dates of stay)
» Reason for Visit*

*We discovered an opportunity around the feature Reason for Visit, which was already being collected in the NAVIS dataset. The options in this features included Family Vacation, Romantic Vacation, etc.

We broke *Reason for Visit* down into two questions for our prototype to better understand the guest's intention for their getaway at Nemacolin. The first question asks who the guest is going with (Partner, Friend, Family, or Other). The second question asks what trip style the guest prefers (Adventure, Romance, Relaxation, Active, Family Fun). The answers to these two questions informed our selection of activities and dining options within the prototype.

2. Incorporate activities and dining experiences into recommendations

Since it was not sufficient to simply recommend a hotel to the guests, we designed the logic to provide recommendations on rooms, activities, and dining based on popularity, relevance, and distance as determined by qualitative research. It is important to note that these recommendations are not made my any Machine Learning model in the prototype.

After deciding on the predictive features to collect from the guest, we began to create the logic to connect selections to the results for stay, activities, and dining.

STAY We relied upon the logic of the Decision Tree to suggest hotel type. Based on hotel type, we selected different rooms listed on the resort's website to include. Our top stay suggestion within the prototype is reflective of the Decision Tree logic. However, the prototype also has similar rooms at other hotels on the property to offer guests variety.

ACTIVITIES Over the past months of qualitative research, the team had an understanding of which activities were thought of as the most popular by the resort. We selected three to five activities for each branch in the logic tree, relying up qualitative popularity, relevance to trip style categories, and the activities listed for the summer season (mid-May through August) on the resort website.

DINING Distance from hotel type determined our suggestions for dining. However, there is an exception for the trip style answer 'Romance'. If a guest selects 'Romance', they will be suggested Lautrec as their first dining option.

3. Refine recommendations through testing

The team checked the branches of our logic tree and the options available to guests through user testing. We asked participants to organize all 75 of the resorts activities into categories, which influenced the trip style options within the prototype. In testing sessions on early versions of the prototype, participants were also asked if the options recommended to them based on trip style seemed relevant. Based on this feedback, we updated our logic to better match guests' recommendation expectations.

Assumptions

Our prototype assumes availability of the activities and dining options recommended to the guest. A production-ready system would take availability into account and only populate offerings that were taking place and had openings during the selected stay dates. We address availability of options within the Itinerary section of the prototype by recommending the guest a time and date for a given experience. When a guest goes to add an activity to their itinerary, the prototype automatically suggests an available time to them. This design not only reduces the effort required of the guest but also allows the resort to reflect availability.

Section IV
Future Implementation Strategy

As mentioned previously, one of the primary limitations of our Machine Learning prototype was that there were not enough relevant features which were readily accessible. The disparate systems currently used by Nemacolin lead to the creation of data silos which prevents the data from being accessed easily. There is no way to link back the purchases of a guest to their reservations at the resort automatically. Having the ability to do so would help understand the most popular items and services offered by Nemacolin. This problem is not unique to Nemacolin. It is in fact a common problem faced by most businesses that use more than one software systems. As a result of the software systems not talking to one another, data needs to be duplicated into different systems. This is often done manually, which introduces the possibility of human error in the duplication process. Additionally, this situation also increases operational costs.

Before a production-ready Machine Learning system can be implemented, Nemacolin should take the following steps:

- **Create a centralized repository of guest information**
  A centralized repository of guest data would allow Nemacolin to find more informative insights as well as allow Nemacolin to leverage Machine Learning. It also resolves the issue of having data silos which prevent information from being readily accessed. We recommend that this issue be one of the first to be addressed.
- **Collect more information on the guests**
  Information about guest motivations and behaviors can help improve conversions as well as allow Nemacolin to do targeted marketing. Collecting more information on guest behavior on the website can help improve what content in shown to the guests thereby improving conversions. This can be achieved with the help of analytics libraries such as Firebase and Google Analytics. These libraries are discussed in more detail later on.

  The Personix data was a good first attempt at collecting psychographic profiles on guests. Acxiom, the company that provides Personix data, offers services to generate detailed psychographic features about customers. This information has the potential to help train Machine Learning models which have very high accuracy.

Our prototype showcases the value of providing guests with a curated and personalized selection of options to choose from. Implementing a production-ready system would provide Nemacolin an opportunity to collect guest information (such as analytical data and purchase preferences). This data can then be used to improve the performance of the system and provide guests with more granular recommendations. The way in which this data could be collected is elaborated upon later in this section.

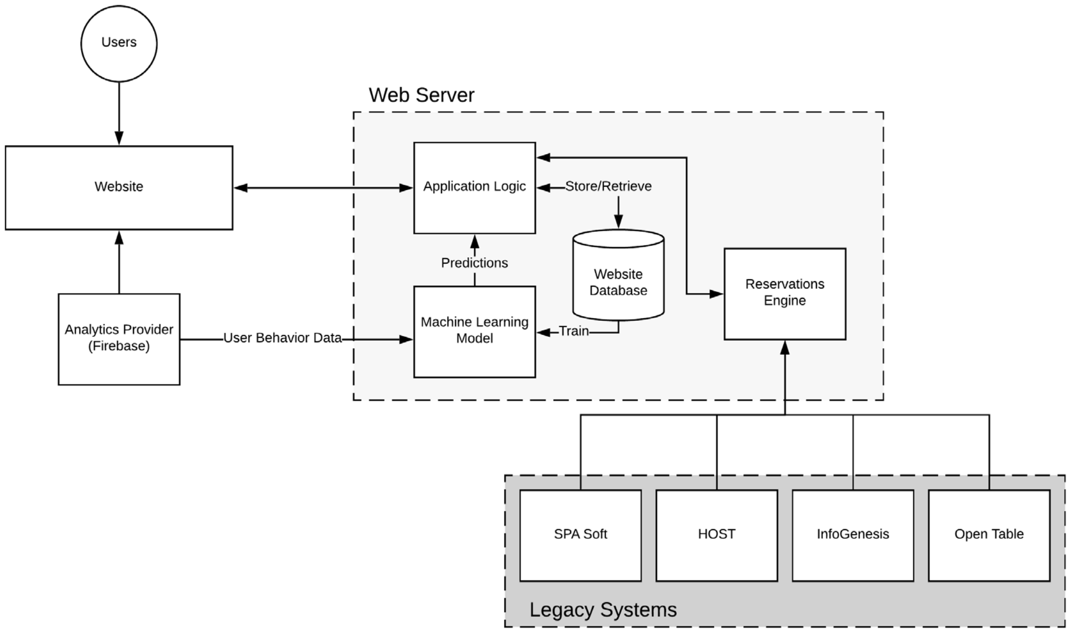


Figure 5: A suggested architecture for the concept web app

Figure 5 proposes an architecture for our concept. It shows how the Web Server for our concept would interact with the legacy systems as well as Nemacolin's website. Once the users are funneled to the new platform, we can collect more information about their behavior and preferences through the use of an analytics platform such as Firebase. We believe that this behavioral data would serve two purposes — to provide insight into how the design of the website could be improved, and reveal patterns in user behavior that will serve as features to train Machine Learning models.

Since the concept is yet another software system, it is crucial for it to have interfaces that allow communication with the existing legacy systems so that the issue of data silos is not exacerbated. Additionally, this web-based solution would also be a good central repository to collect information about potential guests and their behaviors. The centralized repository of data can be used to train recommender systems that provide the guests with the curated selection of options to choose from.

Analytics

Nemacolin currently uses Google Analytics on their website. It is a part of an extremely powerful suite of marketing tools developed by Google. As a result, it is primarily geared towards advertising and business purposes.

We recommend that Nemacolin also implement Firebase alongside Google Analytics. Firebase is a development platform that is owned by Google. There are a few notable features in Firebase that we believe would be very useful to Nemacolin:

» Firebase provides unlimited event logging which would be useful in analyzing guest behavior patterns when they are browsing through Nemacolin's website.

» Firebase helps create dynamic user groups based on their predicted behavior. This feature can be used to customize the experience on the website and increase the conversion ratio for potential guests.

» Firebase integrates with Google Marketing Platform. This allows Nemacolin to leverage the capabilities of both Google Analytics as well as Firebase to increase guest engagement on their website.

» Firebase is going to be especially helpful if the team's prototype were to be in into production. It would help record user behavior and dynamically segment them to provide a more personalized experience.

Data Collection

A recommender system that can recommend between different dining, activity and lodging options should have access to at least the pieces of information shown in **Figure 6**. The diagram shows a possible method to structure the databases which min-

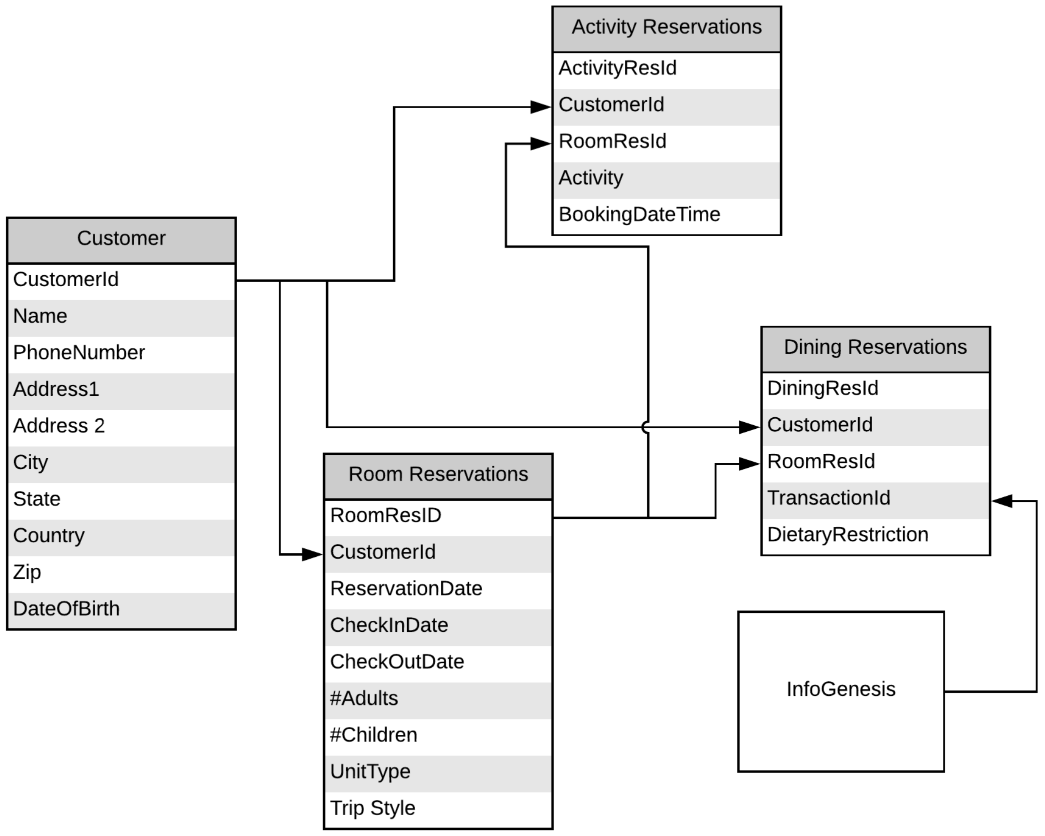


Figure 6: A suggested database schema for collecting guest information

imizes data redundancy. Redundancy tends to increase the size of the data and consequently the costs.

Since the current software systems do not communicate with each other, the web application could also serve as a central repository for all of the guest information. However, for this to be possible, the following steps have to be taken:

- » The data collected by the web application should be interfaced with reservations dataset provided by the HOST system.
- » The research conducted by the team showed that it is currently not possible to link individual purchases made by guests at the resort to their reservations. Building a tool that allows InfoGenesis to interface with the web application can solve this issue.
- » For ensuring that availability is taken into account when making a reservation, the web application also needs to be interfaced with SpaSoft and OpenTable.

In addition to these data points, the ensemble Machine Learning models would also benefit from psychographic, geographical and behavioral information on the guests.

Conclusion

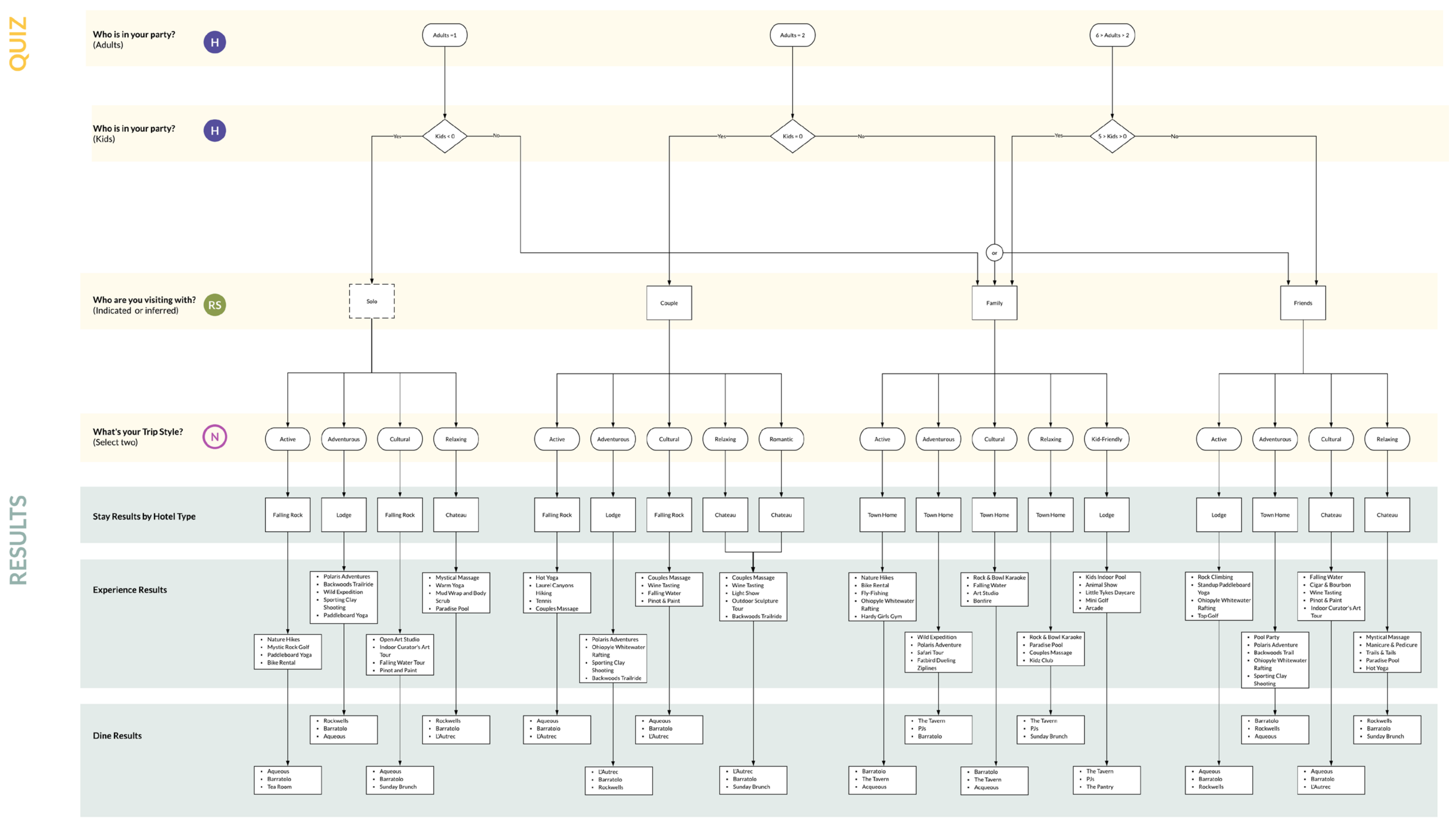
This report presented the findings of the teams exploration into the viability of Machine Learning being used in the design solution. The exploration revealed that while using Machine Learning at Nemacolin could have a positive impact, the current infrastructure at the resort presents a challenge in doing so.

A complete overhaul of the legacy systems is an unrealistic expectation. The team instead recommends that the final design solution be interfaced with the legacy systems and be used as a centralized repository for gathering all guest information. This centralized repository would be tremendously helpful in building automated systems at Nemacolin.

Finally, the report also lists some sources for collecting information on the guests along with the kinds of information that should be collected. This includes demographic, geographic and psychographic data.

Appendix

Logic Flowchart


QUIZ
RESULTS
Who is in your party? (Adults) H
Who is in your party? (Kids) H
Who are you visiting with? (Indicated or inferred) RS
What's your Trip Style? (Select two) N
Stay Results by Hotel Type
Experience Results
Dine Results
Adults =1
Adults = 2
6 > Adults > 2
Kids < 0
Kids = 0
5 > Kids > 0
Yes
No
or
Solo
Couple
Family
Friends
Active
Adventurous
Cultural
Relaxing
Romantic
Kid-Friendly
Falling Rock
Lodge
Chateau
Town Home
Nature Hikes • Mystic Rock Golf • Paddleboard Yoga • Bike Rental
Polaris Adventures • Backwoods Trailride • Wild Expedition • Sporting Clay Shooting • Paddleboard Yoga
Open Art Studio • Indoor Curator's Art Tour • Falling Water Tour • Pinot and Paint
Mystical Massage • Warm Yoga • Mud Wrap and Body Scrub • Paradise Pool
Hot Yoga • Laurel Canyons Hiking • Tennis • Couples Massage
Polaris Adventures • Ohiopyle Whitewater Rafting • Sporting Clay Shooting • Backwoods Trailride
Couples Massage • Wine Tasting • Falling Water • Pinot & Paint
Couples Massage • Wine Tasting • Light Show • Outdoor Sculpture Tour • Backwoods Trailride
Nature Hikes • Bike Rental • Fly-Fishing • Ohiopyle Whitewater Rafting • Hardy Girls Gym
Wild Expedition • Polaris Adventure • Safari Tour • Fatbird Dueling Ziplines
Rock & Bowl Karaoke • Falling Water • Art Studio • Bonfire
Rock & Bowl Karaoke • Paradise Pool • Couples Massage • Kidz Club
Kids Indoor Pool • Animal Show • Little Tykes Daycare • Mini Golf • Arcade
Rock Climbing • Standup Paddleboard Yoga • Ohiopyle Whitewater Rafting • Top Golf
Pool Party • Polaris Adventure • Backwoods Trail • Ohiopyle Whitewater Rafting • Sporting Clay Shooting
Falling Water • Cigar & Bourbon • Wine Tasting • Pinot & Paint • Indoor Curator's Art Tour
Mystical Massage • Manicure & Pedicure • Trails & Tails • Paradise Pool • Hot Yoga
Aqueous • Barratolo • Tea Room
Rockwells • Barratolo • Aqueous
Aqueous • Barratolo • Sunday Brunch
Rockwells • Barratolo • L'Autrec
Aqueous • Barratolo • L'Autrec
L'Autrec • Barratolo • Rockwells
Aqueous • Barratolo • L'Autrec
L'Autrec • Barratolo • Sunday Brunch
Barratolo • The Tavern • Acqueous
The Tavern • PJs • Barratolo
Barratolo • The Tavern • Acqueous
The Tavern • PJs • Sunday Brunch
The Tavern • PJs • The Pantry
Aqueous • Barratolo • Rockwells
Barratolo • Rockwells • Aqueous
Aqueous • Barratolo • L'Autrec
Rockwells • Barratolo • Sunday Brunch

Legend for Data Collection Systems

H HOST database

RS Reservation Specialist calls

N NAVIS database
partially collected as Reason for Stay