

Paper 126-2012

Developing a Predictive Model for Customer Trip Purpose to Be Integrated into Enterprise Strategy and Analytics

Jonathan Levine, Marriott International, Bethesda, MD

ABSTRACT

Marriott International™ needed a single accepted metric of leisure travel as a component of total business that allowed for nuanced measures and the analysis of the impact of specific marketing campaigns. The Marriott Rewards Customer Knowledge team created a predictive model that estimated the likelihood that a stay was for business. The results of this model will be used both in enterprise-wide reporting visible at the C-level as well as in targeted marketing campaigns. This paper will address some of the modeling approaches used, as well as some of the challenges faced, including specific issues related to building a model to be incorporated into a production ETL.

Software used included SAS/STAT®, SAS® Enterprise Miner®, SAS/ACCESS® Interface to Netezza, and SAS® Scoring Accelerator for Netezza.

INTRODUCTION

Marriott International, Inc. (NYSE: MAR) is a leading lodging company based in Bethesda, Maryland, USA with more than 3,700 properties in 73 countries and territories and reported revenues of over \$12 billion in fiscal year 2011. Marriott International has set the strategic goal of increasing its share of the leisure travel market. Being a data-driven company, it was essential to create metrics to measure Marriott's overall volume of leisure travel as well as to assess the impact of programs implemented to achieve this goal.

At the micro level, Marriott's targeted marketing programs use customer stay behavior in order to personalize communications and promotions to increase engagement and drive revenue. Knowing which customers have had leisure travel, as well as which stays were part of leisure trips, would greatly increase the precision of these efforts and more effectively drive revenue.

This required the identification of leisure stays within our reservation database. Unfortunately, there is not a single clear definition of how to identify whether a particular customer stay was for business or leisure. Customers are given the opportunity to state their trip purpose when booking through certain channels, but this is voluntary and not always done. Trip purpose is not available at all for other major booking channels. Therefore, rules of thumb were generally used to identify leisure stays. Such rules might be based on distinguishing weekday vs. weekend days stayed, looking at the rate used, or making separate assumptions for resort stays.

While these approaches were generally acceptable, they captured neither the nuance of customer behavior nor the wide range of properties and brands in the Marriott portfolio. While in general weekday stays at most Marriott International properties are for business travel, customers regularly stay for leisure during the week. Likewise, there are no policies preventing someone from using a discount rate for business when available, nor can it be certain that someone paying the full rate is not a leisure traveler. Lastly, certain properties break from the conventional assumptions about their business patterns, whether it is the resort that also serves as a business hub or the non-resort in or near a leisure destination. A more detailed approach was needed to identify the customer's trip purpose on any given reservation.

MODELING TEAM

The predictive model was built by the Customer Knowledge team within Marriott Rewards, primarily by the author of this paper. Support was given by business partners from other departments to help define and review assumptions. Including business partners early in the process was important for helping to build credibility of the final results. In addition, the business partners would be some of the main users of the results of the model and helped to define how its output would be included in strategic analyses.

GENERAL APPROACH

The approach chosen was to use data mining techniques to identify the patterns within reservation data that could be generalized to estimate customers' trip purpose on all reservations. The training dataset was made up of the records where a customer had stated the trip purpose of his or her stay (business or leisure) via one of two booking channels (Marriott.com and On Property) that were deemed to be accurate enough for training a model. This specification became the binary target variable, and a predictive model was built using SAS Enterprise Miner to estimate the likelihood that a reservation was for business given the data on the reservation record.

The approach was limited by the following factors:

- Trip purpose was only populated on a portion of records, and only from two of our booking channels. An assumption was made that patterns found in these records could be extrapolated to other channels and to records without populated values. The end results applied to all reservations were presented to multiple experts for validation, and the results were judged to be reasonable.
- Due to requirements of the Extract, Transform, and Load (ETL) process that creates the table that will contain the model results, all of the inputs to the final model to be used in scoring had to be available within or derived from a single reservation record. It was not possible to look across reservations when scoring a given reservation.
- Reservations from group events were excluded from this analysis, as they would require an entirely different approach using different data about the entire event and all its participants.

Customer Knowledge met with key stakeholders to understand the issue. There were two analytical goals to this project. The first was to create a macro-level estimate of leisure activity that could be reported at the enterprise level as well as allow for drill-down to at least the hotel property level. The second goal was to create micro-level estimates of trip purpose at the stay level that could be used to model the behavior of specific customers at specific times.

To achieve both goals, it was decided that we would model trip purpose at the level of the individual reservation and store the results in the Enterprise Data Warehouse as a new field in the detailed reservation analysis table. The new field would be the likelihood that a given reservation had a trip purpose of business. For aggregate reporting, a methodology was developed to summarize the trip purpose of individual reservations into an aggregate value that accounted for the uncertainty in the predictive model. Uncertainty was handled by allocating room nights to business or leisure based on the modeled probability, where in the aggregate value higher probabilities counted more and values closer to 50% were split evenly between business and leisure. This could give an enterprise view of leisure activity, while allowing detailed analysis of specific programs or customer behavior.

INCORPORATING RESULTS INTO DATA WAREHOUSE

An important aspect of this project was incorporating the results into the enterprise data warehouse. This allows the results to be used by all analysts across the enterprise, either by record-level analysis or by reporting via business intelligence tools. Marriott's data warehouse has business intelligence software layered on top of it. Power users are able to access record-level data via SQL, whereas enterprise reporting is handled through reports and dashboards.

In order to generate model scores faster and more efficiently, Marriott recently acquired the SAS Scoring Accelerator for Netezza™. This allows predictive models built in SAS Enterprise Miner™ to be translated into in-database functions within our Netezza Data Warehouse. The model will be turned into a scoring function, and this function will be used in the ETL of the reservations data analysis table that is used by power analysts, as well as in general reports.

At the inception of this project we had planned to purchase the software, but the purchase and installation had not been completed. This delayed some of the implementation, but was seen as a best approach rather than having IT manually code the model into their ETL.

Using the scoring function enables the modeling process to be separated from the scoring and ETL process. The modeling team will provide the function call to the IT department that is responsible for the ETL. IT will, in turn, incorporate the results as a new column in the analysis table in the data warehouse and apply calculations to use the results on all derived aggregate tables. Separating the modeling process from the scoring ETL achieved three major goals:

1. The modeling team is not required to become directly involved in the scoring process. If the model had been built using only base SAS code, then the modeling team would have to provide detailed specifications for how to score the data, including all transformations and calculations. Instead, the modeling team will just have to provide the function call which only uses inputs from the reservation table and one supplemental table. IT can just call this function and insert the results into a new column in the table.
2. Without the Scoring Accelerator, either the modeling team would have been limited to functions that could be coded in SQL, or the IT team would have had to download the entire table containing all of Marriott's reservation data into SAS in order to score it. The results would then be uploaded back into the data warehouse and matched to the original data. This would be a significant burden for the IT team both in terms of development time and production run-time. In addition, the process of extracting data from the data warehouse, processing it, and then updating it within the data warehouse would have created multiple opportunities for process failure, data integrity problems, and security risk.
3. By creating a scoring function, the modeling team has the flexibility to change the model at a later date without forcing the re-development of the ETL. In the future, we can use different modeling techniques, add additional transformations, or just refit the model using more recent data. By republishing the model, which is a simple process using Scoring Accelerator, we can change the entire function without creating a new project for our IT partners, assuming the input parameters to the function do not change.

While there were great advantages to using the Scoring Accelerator, using the product did cause some challenges. This was a new product for Marriott, so we were learning as we went how to best build a model using the product. There were several times where the overall modeling approach or specific aspects of the code had to be changed in order to optimize the use of the product. These issues will be discussed below.

MODELING ISSUES

Building a predictive model of customer trip purpose raised a number of analytical challenges on top of the normal issues in predictive modeling. The rest of this paper will describe each of these challenges and how they were resolved.

DIFFERENT TYPES OF STAYS (STAY PATTERN)

Customers have many different types of stays with Marriott, and the input variables might have differing relationships with the target depending on the type of stay. The typical business traveler stays during the week and weekends are often for leisure. Some stays can last months or even years¹.

As an example, the rate paid might have a significant correlation with business travel for a weekday stay, but could be irrelevant for a weekend stay and actually have the opposite relationship for an extended stay. In addition, rates available to a customer differ depending on the type of stay, so the different stay patterns must be isolated to bring out these different effects.

¹ <http://www.blogs.marriott.com/marriott-on-the-move/2012/01/our-10-year-guest.html>

In the end, we settled on four stay patterns, based on arrival day, departure day, and length of stay:

- Weekday
- Weekend
- Extended Stay
- Combination (Weekday and Weekend nights on the same stay)

The specific definitions and number of stay patterns changed over the modeling process as we considered borderline cases.

LARGE NUMBER OF CATEGORICAL INPUT VARIABLES

Traditional predictive modeling methods such as regression require numeric inputs in order to predict the target variable. The majority of data available on a reservation record, however, is categorical. Fields such as arrival and departure day, property, rate classification, time of year, etc. are all key inputs but must be translated into numeric variables in order to be used.

The approach taken was to create a set of interval variables scaled between 0 and 1 which represented the correlation of these categorical variables with the target variable. These interval variables were termed “factor” variables. For example, the “rate type code field,” which contains general groupings of rate codes, was translated from values such as “Business”, “Discount”, and “Wholesale” into a numeric field ranging from 0 to 1, with the leisure rates having a value closer to 0 and the business rates having a value closer to 1. The factor variables were created by building a series of predictive models. Each model used a small set of related categorical inputs in order to isolate the effect of that input on the target variable. This method successfully transformed unordered categorical variables into a numeric value that was meaningful both in terms of order and scale and thus could be used as an input to the final model.

The most challenging factor variable was the one for specific property. Each property has its own characteristics as to whether it is primarily a leisure, business, or mixed destination. Many fit general categories, such as resorts, but there are many exceptions. Our Courtyard by Marriott™ brand is targeted towards business travelers, but we also have Courtyard hotels near major vacation destinations that may in fact be primarily used for leisure. On the other hand, some resorts serve as business hubs in their markets. Business travelers going to a Caribbean island, for instance, may still stay at a resort as there are not many typical business transient hotels. Initially, the business percent of the hotel was used as an input, but that proved to be too granular. In the final model, a decision tree was used to bin hotels into 10 groupings based on their overall leisure component. In all of these cases, the factors were derived separately for each stay pattern.

MIX OF DATA-DRIVEN AND SME-DETERMINED SEGMENTATION

Segmentation is an important step of the modeling process. There are two major reasons to segment the data when fitting regression-type models (including neural networks).

- Different segments may have different relationships between the input and target variables or have different variables that are important. Without segmentation, the effects would be blurred or key variables might be lost in the variable selection process. Segments such as these are best identified using the judgment of subject matter experts.
- Distinct segments have higher or lower intercept values, and not segmenting allows the largest segment to dominate the average and thus hide the smaller but often important segments. Segmentation allows for a de-averaging of the target value that leads to better model accuracy. This segmentation can best be determined systematically.

Both types of segmentation were relevant to this model. As mentioned earlier, the stay patterns have different characteristics and require separate models. However, within the stay patterns there are also distinct segments that are not so easily identified a priori by subject matter experts. For instance, both a 10-night stay at a resort and a 30-night stay at one of our Residence Inn by Marriott™ properties can be considered an “extended stay”, but the first is likely to be leisure and the latter is likely to be business. Likewise, other factors such as specific property characteristics, rate paid, seasonality, etc. may be important, and it is best to use a pattern finding approach such as a decision tree to identify key segments before fitting a final model.

Unfortunately, it is difficult within Enterprise Miner to apply both methods in a single modeling path. It would require first applying the business rules to define the stay patterns, then using the Start Group/End Group nodes to fit separate decision trees to create segments within each stay pattern. So far, this is fine. However, to then apply variable selection and fit models for each stay pattern and segment would require a second set of Start Group/End Group nodes, either nested or in serial. Two methods were initially considered which would allow this type of double segmentation within Enterprise Miner, but neither is permitted by the tool. The first is to nest the group processing loops as shown in the Enterprise Miner diagram in figure 1:



Figure 1 - Nested Group Nodes (not permitted)

The second is to first use a loop to create different segments within each stay pattern, and then add another loop, in serial, that loops based on the stay pattern and segments, as shown in the Enterprise Miner diagram in figure 2:



Figure 2 - Serial Group Nodes (not permitted)

However, neither is permitted by Enterprise Miner 7.1. The prior will not run, and the latter will run but the score code will generate errors.

Moreover, the models for deriving the “factor” variables from nominal inputs also needed to be fit at the stay pattern level. The solution was to do the transformation and segmentation in Base SAS using the Enterprise Miner Procedures PROC DMREG and PROC ARBOR. Each one generated score code that could be pasted into a Transform Variables node. The final approach is described in detail below. Example code can be found in the appendix to this paper. The approach used is shown in the Enterprise Guide SAS code and the Enterprise Miner diagram in figure 3.

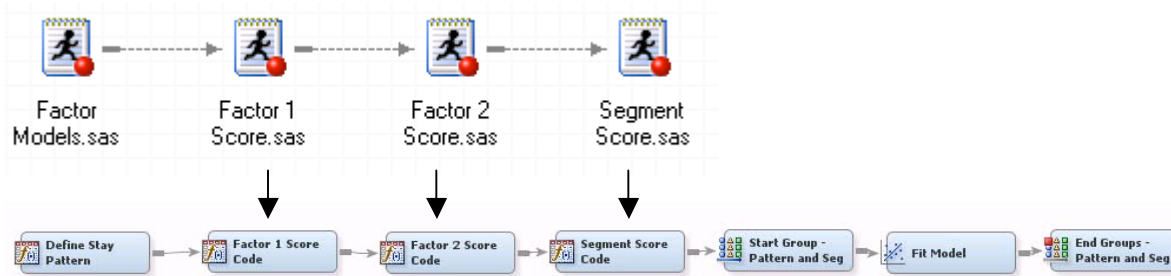


Figure 3 - Factor Score Code from Base SAS Applied via Transform Variable Nodes (Used in Model)

MODELING PROCESS

After the definition of the problem and initial data exploration were complete, the final modeling approach was as follows:

1. Build modeling dataset from reservation data
2. Apply initial transformations to derive new variables
3. Fit models to transform nominal inputs into interval inputs by creation of “factor” variables
4. Use Transform Node within Enterprise Miner to score inputs to generate each factor
5. Fit final models for each stay pattern/segment
6. Score within Enterprise Miner and export model to Netezza
7. Validate results of model
8. Deploy model for analytics and reporting

STEP 1 – BUILD MODELING DATASET FROM RESERVATION DATA

It is very important to keep in mind the desired usage of the model when building the modeling dataset. In our case, the goal was to create a scoring function that would use untransformed columns from our reservation data as inputs to the model. Fields such as “Arrival Date,” “Departure Date,” “Property,” and so on appeared on the reservation data. SAS datasets were created by accessing the reservation data using SAS/Access for Netezza to download the table to the analytical workstation. The fields were not transformed from the reservation data. The records that would be used in the modeling process were broken out into training, validation, and test datasets. Reservations without a stated trip purpose were not used for modeling, but were set aside for subject matter expert validation and for presentation of results.

Since some of the steps were to be performed in Base SAS, the datasets were partitioned into training, validation, and test data before importing into Enterprise Miner, rather than using the Partition node. That way, the models for the various “factor” variables could be fit using the same training data as the final model, avoiding a contamination of the model validation process.

This required creation of three separate Data Input nodes within Enterprise Miner, one for each dataset with the respective roles of train, validate, and test. The first step of the model was to link them all to a control point so that all subsequent steps would be performed on each dataset.

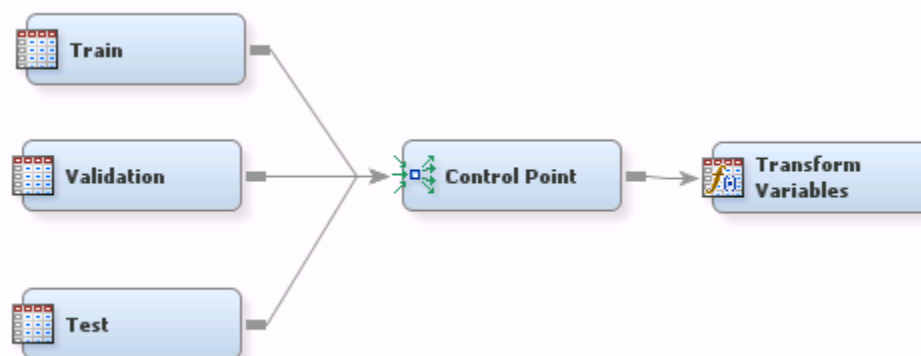


Figure 4 - Input Data

Given that a goal of this process was to create a scoring function that would be incorporated into an ETL while maintaining flexibility for future model updates, two principles were followed:

- All input variables to be called by the function existed on the main database table to be scored. Transformations were applied within the scoring function, rather than asking IT to apply those transformations and then call the function with the transformed variable. This made it simpler for IT while also giving maximum flexibility in case new or different transformations would be used later.
- In some cases, variables were included that might be used in the future, even if they were not used in this model. That way, future models could incorporate those variables by just updating the function, and without requiring new development by IT to incorporate a new parameter into the function.

STEP 2 – APPLY INITIAL TRANSFORMATIONS TO DERIVE NEW VARIABLES

While SAS Enterprise Miner is a powerful tool for fitting predictive models, it still relies on the subject matter expertise of the predictive modeler. In this case, the data on the reservations records was transformed into variables that could be used as inputs to a predictive model. For instance, fields such as arrival date, departure date, and booking date could not be used raw, but had to be interpreted and their key elements brought to the fore.

Some of the main transformations were:

1. The day of week and month of the year were extracted from the arrival/departure dates to calculate the following inputs to the model:
 - a. Arrival day of week
 - b. Arrival month (for seasonality)
 - c. Stay pattern (weekday, weekend, etc.)
 - d. Length of stay
2. Property geography and brand were aggregated from the more granular fields in the data.
3. Certain text entry fields such as guest company name, credit card type, address type, and so on were parsed to identify particular inputs that were meaningful.

Two of these transformations led to particular challenges when trying to publish the model using Scoring Accelerator. Initially, the SAS `weekday()` and `month()` functions were used to extract the day of week and month of year from dates. However, these functions cannot be used in models that will be scored by Scoring Accelerator 1.6, as the functions have not been ported into Netezza. The end solution was to use SAS standard date formats `downname` and `mmddy` and the `put()` function to extract the day and month.

This step was replicated in the initial transformations in the Base SAS code used for fitting the “factor” models and in the first step of the Enterprise Miner diagram via a Transform Variables node using the “SAS Code” component.

STEP 3 – FIT MODELS TO TRANSFORM NOMINAL INPUTS INTO INTERVAL INPUTS BY CREATION OF “FACTOR” VARIABLES

A key challenge of this model was that the majority of input variables were nominal, whereas the best predictive modeling techniques use interval inputs. Two approaches are commonly used to address this issue. The simplest, implemented by default within Enterprise Miner, is to create a set of dummy indicators. However, in this model, the number of dummy indicators would be excessive and lead to an ill-formed model while dramatically increasing run-time. For instance, the specific property was an input to the model, in order to let the data tell which properties were more oriented towards leisure or business. The Marriott International portfolio of properties is 3700+ and growing, so this would be too difficult to model. Other key variables included:

1. Time of year
2. Rate used

3. Property type
4. Room type
5. Arrival day of week

The solution was to fit several predictive models to derive the relationship between one or more of the nominal inputs and the target variable. Separate models were fit for each stay pattern, as the relationships might differ among the stay patterns. A macro was written in Base SAS that looped through each of the stay patterns and fit eight different models, each model using one to three related input variables. The models were fit using the Enterprise Miner Procedures that may be called from Base SAS code when Enterprise Miner is licensed and installed.

For cases where the nominal variables took a small number of distinct values, PROC DMREG was used to fit a logistic regression with interaction effects. For the cases where the nominal variables had many distinct values, PROC ARBOR was used to fit a decision tree with maximum width of 10 leaves and depth of 1 layer to group the values into bins based on their relationship with the target variable. Each of these procedures has the option to generate SAS scoring code, and this code was saved for the final step.

The next step performed for each stay pattern was to score the original training and validation data with the score code generated above to create eight new factor variables. This was done in a single data step, where the score code was linked via an %INCLUDE statement, and then the final predicted value was renamed to the name of a given factor, such as property_type_factor. Once the data was scored, a new decision tree was built with PROC ARBOR to separate the records into segments with differentiated values for likelihood of business. This decision tree was intentionally kept small (no more than four leaves) so that individual models could be fit for each segment. The score code was also saved.

The final macro compiled score code for each factor variable across all of the stay patterns. This macro read in the individual score.sas files generated by the procedures and wrote out a single .sas file with scoring code. The macro wrapped the individual score code for each pattern in “if-then-else” statements so that the final file would be able to score any record, as long as the stay pattern was available. Finally, it appended code to rename the predicted value with the factor name and to drop all of the special variables generated by Enterprise Miner.

An example of this code can be found in the appendix to this paper.

STEP 4 – USE TRANSFORM NODE WITHIN ENTERPRISE MINER TO SCORE INPUTS TO GENERATE EACH FACTOR

At this point, the Enterprise Miner UI was used to fit the final model. After a Transform Variables node was used to apply initial transformations, nine additional Transform Variable nodes were used to create each of the eight factor variables and the model segment. The score code generated by the macro described above was copied and pasted into the “SAS Code” portion of each Transform Variables node.

For instance, given that the target variable was named “target_purpose_business_ind,” the score code for the “rate factor,” with separate models fit for each stay pattern, might look like the following:

```
if stay_pattern='Weekday' then do;
  [PROC DMREG score code for weekday pattern]
end;
else if stay_pattern='Weekend' then do;
  [PROC DMREG score code for weekend pattern]
end;
else if stay_pattern='Extended' then do;
  [PROC DMREG score code for extended stay pattern]
end;
else if stay_pattern='Combination' then do;
  [PROC DMREG score code for combination pattern]
```



```

end;

rate_factor=p_target_purpose_business_ind1;

drop i_target_purpose_business_ind _node_ _leaf_ p_target_purpose_business_ind1
p_target_purpose_business_ind0
q_target_purpose_business_ind1 q_target_purpose_business_ind0
v_target_purpose_business_ind1 v_target_purpose_business_ind0
u_target_purpose_business_ind _warn;

```

This process was necessary to allow the models to be fit separately for each stay pattern, but to be treated by Enterprise Miner as if the results were just like any other interval input variables or nominal segment variables. This allowed group processing to be used later in the process flow, thus solving the problem described earlier concerning having two group processing loops in the same process flow. The macro that generated this score code can be found in the appendix to this paper.

STEP 5 – FIT FINAL MODELS FOR EACH STAY PATTERN/SEGMENT

The prior steps created all of the input variables that could potentially be used within the final predictive models. The next step was to add a “Metadata” node to drop the unnecessary variables, including the original nominal input variables, and to make sure the new variables had the right role and type.

Then, group processing in Enterprise Miner (Start/End Group Nodes) was used to loop through each combination of Stay Pattern and Segment. For each segment, two Neural Networks were fit. The difference between the two was the variable selection method used – stepwise logistic regression or the Variable Clustering node. Within the loop, a Model Comparison node was used to determine the best model, using the ROC metric on the validation data.

Each Neural Network was a standard Multilayer Perceptron Neural Network with 10 nodes in the hidden layer. All had the same target – a binary indicator of whether the stay was for business or leisure.

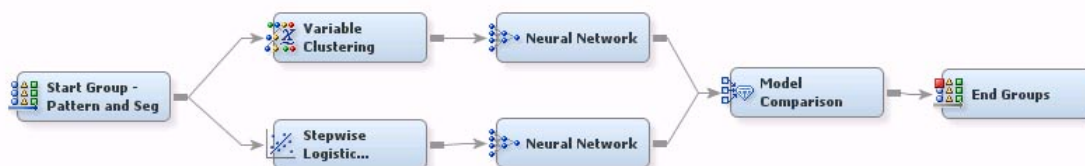


Figure 5 - Model Fitting Loop

STEP 6 – SCORE WITHIN ENTERPRISE MINER AND EXPORT MODEL TO NETEZZA

The Enterprise Miner Score Node was then used to create score code that could be used for several purposes. First, the test data was scored and analyzed from a number of perspectives to measure model accuracy. Second, all of the reservation data was then scored, even if the actual trip purpose was not known, including for booking channels outside the scope of the model. This allowed subject matter experts to validate the results based on their experience and to report results.

Then the “Score Export” node was used to create the files that are needed as an input to the Scoring Accelerator for Netezza™. The Score Export node creates score code and an XML file with the necessary information to create a scoring function. This function was then deployed to the Netezza Data Warehouse and is available for use by analysts.

STEP 7 – VALIDATE RESULTS OF MODEL

The model was validated in two complementary ways. First, a test dataset had been held out from the initial data and was used to assess the accuracy of the model against data for which the true purpose was known. Some key metrics used to validate the model were as follows:

- Using a cutoff of 50% probability for assigning the model's predicted trip purpose, the model was found to accurately predict purpose for 87% of records. This was compared to accuracy rates of 60-70% for previously used heuristics and rules of thumb.
- Since this model will be used to assess the success at individual properties, it is important that the results be valid at a more granular level. Therefore, the overall predicted business percent at the property level was compared to the actual business percent and found to be within 10% of actual for all but a handful of properties.
- Model accuracy was analyzed at various other levels of detail to ensure there were no major anomalies in the predictive value. The model accuracy was found to be consistently high across all rate types, all property types, and across different times of the year.
- A chart was created comparing the predicted vs. actual percentage of business for different binned values of modeled probability. For instance, for records where the model predicted 5% likelihood that the reservation was for business, the actual percentage was close to 5%. As the prediction increased, the actual value increased in parallel. This shows that the model was both accurate in its predictions and also properly reflected when it was less confident in its results.
- The large majority of records had either a very low (<20%) or very high (>80%) modeled probability of business. Thus the model's predictions were useful in the sense that they clearly separated business from the leisure stays, while allowing for uncertainty when necessary.

Afterwards, the entire universe of reservations was scored against the model. These included the reservations for which the actual trip purpose was not known, and thus model accuracy could not be directly assessed. Instead, summaries were presented to business experts who validated that the results made sense in the context of the business. These summaries were also used to drive strategic insights and re-consider strategic initiatives.

STEP 8 – DEPLOY MODEL FOR ANALYTICS AND REPORTING

Deployment of the model is still outstanding as of the completion of this paper. Due to some internal delays in acquiring and deploying the Scoring Accelerator, Marriott has not implemented the model in production. However, it has been deployed and tested in the development environment. Once the deployment is complete, the scoring function will be available to be used by analysts.

While the Scoring Accelerator is significantly faster than downloading results and scoring in SAS, there is still some processing time involved to execute the function in a SQL query. In order to reduce the load on the system during normal hours, the scoring function will be applied during the ETL of the analytical table, with the results available to all analysts with access to the data. The model prediction will be a new column on the table. Therefore, analysts will be able to use the model by taking advantage of the analytical capabilities of the data warehouse without having to execute the scoring function directly, thus reducing the workload on the system during business hours.

In addition, aggregate tables and reports derived from this table will be able to include summaries of leisure and business activity derived from this score. These will be accessed by business intelligence tools to create reports and dashboards.

CONCLUSIONS

This successful project brought a number of benefits to Marriott International:

1. The results produced by the model gave senior management new insights about the nature of Marriott's leisure business, and helped drive strategic decisions about how to increase that business.
2. The ability to have an assessment of trip purpose at the level of the customer stay created the potential to dramatically improve Marriott's targeted marketing programs, as separate response models for business and leisure travel can be developed that considers the nature of past stays.
3. The Customer Knowledge team used this project as a vehicle to increase its understanding of several SAS products, including Enterprise Miner and Scoring Accelerator, which will be leveraged in future projects.

The creation of a trip purpose model was a successful project on a number of fronts and was a great example of applying highly technical analytical methods to enterprise business strategy.

ACKNOWLEDGEMENTS

I would like to acknowledge my supervisor, Stephan Chase, Vice President of Customer Knowledge. He gave me strategic guidance throughout the modeling process as well as invaluable help in learning how to present my results in a way that was meaningful to the enterprise. He has also been supportive in my presentation of this paper.

I would also like to thank Wayne Thompson, product manager for SAS Predictive Analytics, and Jonathan Wexler, product manager for SAS High-Performance Data Mining and In-Database Analytics, for their helpful comments on this paper as well as insights into the product that helped me to solve some modeling challenges.

Lastly, I would like to thank my wife, Risa Weinstein, for her help with the editing of this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jonathan Levine
Marriott International
10400 Fernwood Rd 55/9MR.03
Bethesda, MD 20817
jonathan.levine2@marriott.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX – SAS CODE FOR FITTING FACTOR MODELS

```

***Note: This code requires that SAS Enterprise Miner be licensed and installed;

**declare a list of the four stay patterns to be used in macro loops;
%let pattern_list=WKDY WKND EXTD CMBO;

**declare the location where the score code files should be placed;
%let score_path=F:\Trip Purpose Model\Score Code - Preliminary Input Models;

*****;
*** Step 1: Fit Predictive Models to derive factor variables ****;
*****;

%macro stay_pattern_loop;

**loop through each stay pattern and fit separate models for each;

%do i=1 %to 4;

%let pattern=%scan(&pattern_list.,&i.);

***build dtree and regression models for different nominal variables;
***start by creating dataset with nominal variables and building dmdb;

***select just the records for the current stay pattern;
***source datasets should be sorted and indexed on stay_pattern for efficiency;

data mdl_tmp.pattern_data_train;
  set mdl_tmp.model_data_train;
  where stay_pattern="&pattern.";

  **only keep target and inputs to be used for models;
  keep target_purpose_business_ind rate_type_cd property_cd;
run;

data mdl_tmp.pattern_data_validate;
  set mdl_tmp.model_data_validate;
  where stay_pattern="&pattern.";
  keep target_purpose_business_ind rate_type_cd property_cd;
run;

** PROC DMREG requires the creation of a data modeling database with PROC DMDB;

proc dmdb data=mdl_tmp.pattern_data_train
          out=mdl_tmp.pattern_data_train_dmdb
          dmdbcat=mdl_tmp.dmdb_cat batch;
  class target_purpose_business_ind (desc) rate_type_cd property_cd;
  target target_purpose_business_ind;
run;

proc dmdb data=mdl_tmp.pattern_data_validate
          out=mdl_tmp.pattern_data_validate_dmdb
          dmdbcat=mdl_tmp.dmdb_cat batch;
  class target_purpose_business_ind (desc) rate_type_cd property_cd;
  target target_purpose_business_ind;
run;

***Model 1 - Property Specific Factor;

***example of decision tree to bin properties using PROC ARBOR;

```

```

***declare the fileref for the pattern-specific score code for model;
***PROC ARBOR also creates a "description" file that is useful to keep;

filename scr_prop "&score_path.\Score - Property Specific Factor - &pattern..sas";
filename dsc "&score_path.\Description - Property Specific Factor - &pattern..txt";

proc arbor data=mdl_tmp.pattern_data_train
    leafsize=5000
        criterion=entropy
        maxbranch=10
        maxdepth=1
        missing=bigbranch;
    target target_purpose_business_ind /level=nominal;
    input property_cd /level=nominal;
    assess validate=mdl_tmp.pattern_data_validate;
    describe file=dsc;
    code file=scr_prop leafid prediction noresidual;
run;quit;

***model 2 - Rate Type factor;

***example of regression for nominal inputs using PROC DMREG;

filename scr_rtyp "&score_path.\Score - Rate Type Factor - &pattern..sas";

proc dmreg data=mdl_tmp.pattern_data_train
    dmdbcat=mdl_tmp.dmdb_cat
        validate=mdl_tmp.pattern_data_validate
        outest=mdl_parm.tpm_rate_type_code_&pattern.
        namelen=32
        noprint;
    class target_purpose_business_ind rate_type_cd;
    code file=scr_rtyp group=&pattern._rtyp;
    model target_purpose_business_ind = rate_type_cd
        /selection=none;
    performance cpucount=12 threads;
run;

***delete data;

proc datasets library=mdl_tmp nolist;
    delete pattern_data_train pattern_data_validate;
quit;

***score data against each model to transform nominal variables to interval;
***keep relevant variables;
***dataset will then be segmented based on factor variables;

data mdl_tmp.pattern_data_train;
    set mdl_tmp.model_data_train;
    where stay_pattern="&pattern.";

    %include scr_prop;
    property_specific_factor=p_target_purpose_business_ind1;

    %include scr_rtyp;
    rate_type_code_factor=p_target_purpose_business_ind1;

    keep target_purpose_business_ind property_specific_factor rate_type_code_factor;

run;

data mdl_tmp.pattern_data_validate;
    set mdl_tmp.model_data_validate;

```

```

where stay_pattern="&pattern.";

%include scr_prop;
property_specific_factor=p_target_purpose_business_ind1;

%include scr_rtyp;
rate_type_code_factor=p_target_purpose_business_ind1;

keep target_purpose_business_ind property_specific_factor rate_type_code_factor;

run;

**Example of using PROC ARBOR to create a decision tree for segmenting data;
**Inputs are the various factor variables created;
**At most 4 segments (2 branches x 2 depth) will be created for each stay pattern;

filename scr_seg "&score_path.\Score - Model Segment - &pattern..sas";
filename dsc "&score_path.\Description - Model Segment - &pattern..txt";

proc arbor data=mdl_tmp.pattern_data_train
      leafsize=5000
      criterion=probchisq
      maxbranch=2
      maxdepth=2
      missing=bigbranch;
  target target_purpose_business_ind /level=nominal;
  input  property_specific_factor rate_type_code_factor /level=interval;
  assess validate=mdl_tmp.pattern_data_validate;
  describe file=dsc;
  code file=scr_seg leafid prediction noresidual;
run;quit;

***delete datasets to keep directory clean;

proc datasets library=mdl_tmp nolist;
  delete pattern_data_train pattern_data_validate
         pattern_data_train_dmdb pattern_data_validate_dmdb;
quit;

%end;
%mend stay_pattern_loop;

***execute first macro;
%stay_pattern_loop;

*****
****  Step 2: Create single score code file for each factor variable****
*****

**List, in order, the 4-character alias used in fileref for each factor;
**this should be the same as the fileref used below, such as scr_prop;

%let file_list=prop rtyp seg;

**List of the factor variables to be created, in same order as file_list;

%let var_list=property_specific_factor rate_type_code_factor model_segment;

**Declare fileref for final score code;

filename fin_prop "&score_path.\Score - Property Specific Factor - All
Patterns.sas";
filename fin_rtyp "&score_path.\Score - Rate Type Factor - All Patterns.sas";
filename fin_seg "&score_path.\Score - Model Segment - All Patterns.sas";

```

```

***This macro loops through each factor;
***First, it writes out a header for the file;
***Then it loops through the individual score code for each stay pattern and
inserts it into the final file, wrapped in "if-then-else" code;
***Finally, it writes out a footer for the file with assignment and drop
statements;

%macro score_file_loop;

***first write headers for each score code file;
%do v=1 %to 3;
%let file=%scan(&file_list.,&v.);
%let var=%scan(&var_list.,&v.);

data _null_;
  file fin_&file.;
  put "**** Trip Purpose Model                      ****;";
  put "**** Score Code for &var.                    ****;";
  put "**** To be pasted into Transform Variable Node in Enterprise Miner ****;";
run;
%end;

***loop through four stay patterns, writing out score code for each file;
%do p=1 %to 4;

%let pattern=%scan(&pattern_list.,&p.);

filename scr_prop "&score_path.\Score - Property Specific Factor - &pattern..sas";
filename scr_rtyp "&score_path.\Score - Rate Type Factor - &pattern..sas";
filename scr_seg "&score_path.\Score - Model Segment - &pattern..sas";

%do v=1 %to 3;
%let file=%scan(&file_list.,&v.);
%let var=%scan(&var_list.,&v.);

data _null_;
  file fin_&file. mod;
  put "if stay_pattern='&pattern.' then do;";
run;

data _null_;
  file fin_&file. mod;
  infile scr_&file.;
  input;
  put _infile_;
run;
data _null_;
  file fin_&file. mod;
  put "end;";
  put "else";
run;

%end;

%end;

***write out closing code for each file;
%do v=1 %to 3;
%let file=%scan(&file_list.,&v.);
%let var=%scan(&var_list.,&v.);

data _null_;
  file fin_&file. mod;
  put "do;";

```



```
put "p_target_purpose_business_ind1=.";";
put "end;";
put;
%if &var.=model_segment %then %do;
put "&var.=_node_.";";
%end;
%else %do;
put "p_target_purpose_business_ind1=";";
%end;
put;
put "drop i_target_purpose_business_ind _node_ _leaf_
p_target_purpose_business_ind1 p_target_purpose_business_ind0";
put "      q_target_purpose_business_ind1 q_target_purpose_business_ind0
v_target_purpose_business_ind1 v_target_purpose_business_ind0";
put "      u_target_purpose_business_ind _warn_.";";
run;
%end;

%mend score_file_loop;

%score_file_loop;
```