Cascaded Machine Learning Model for Efficient Hotel Recommendations from Air Travel Bookings

Eoin Thomas*
Antonio Gonzalez Ferrer*
eoin.thomas@amadeus.com
Amadeus SAS
Sophia Antipolis, France

Benoit Lardeux
Amadeus SAS
Sophia Antipolis, France

Mourad Boudia
Amadeus SAS
Sophia Antipolis, France

Christian Haas-Frangii
Amadeus SAS
Sophia Antipolis, France

Rodrigo Acuna Agost
Amadeus SAS
Sophia Antipolis, France

ABSTRACT

Recommending a hotel for vacations or a business trip can be a challenging task due to the large number of alternatives and considerations to take into account. In this study, a recommendation engine is designed to identify relevant hotels based on features of the facilities and the context of the trip via flight information. The system was designed as a cascaded machine learning pipeline, with a model to predict the conversion probability of each hotel and another to predict the conversion of a set of hotels as presented to the traveller. By analysing the feature importance of the model based on sets of hotels, we are able to construct optimal lists of hotels by selecting individual hotels that will maximise the probability of conversion.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**;

KEYWORDS

Recommender systems, machine learning, hotels, conversion.

1 INTRODUCTION

In the United States, the travel industry is estimated to be the third largest industry after the automotive and food sectors and contributes to approximately 5% of the gross domestic product. Travel has experienced rapid growth as users are willing to pay for new experiences, unexpected situations, and moments of meditation [9, 28], while the cost of travel has decreased over time in part due to low cost carriers and the sharing economy. At the same time, traditional travel players such as airlines, hotels, and travel agencies, aim to increase revenue from these activities. The supply side must identify its market segments, create the respective products with the right features and prices, and it has to find a distribution channel. The traveller has to find the right product, its conditions, its price and how and where to buy it. In fact, the vast quantity of information available to the users makes this selection more challenging.

Finding the best alternative can become a complicated and time-consuming process. Consumers used to rely mostly on recommendations from other people by word of mouth, known products from advertisements [20] or inform themselves by reading reviews [6, 18]. However, the Internet has overtaken word of mouth as the primary medium for choosing destinations [23] by guiding the user in a personalized way to interesting or useful products from a large space of possible options.

Many players have emerged in the past decades mediating the communication between the consumers and the suppliers. One type of player is the Global Distribution System (GDS), which allows customer-facing travel agencies (online or physical) to search and book content from most airlines and hotels. Increased conversion is a benefical goal for the supplier and broker as it implies more revenue for a lower cost of operation, and for the traveller, as it implies quicker decision making and thus less time spent on search and shopping activities.

In this study, we aim to increase the conversion rate for hospitality recommendations after users book air travel. In Section 2, the problem is formulated in order to highlight the considerations which separate this work from many recommender system paradigms. Section 3 presents the main techniques and concepts used in this study. In Section 4, a brief overview is given of the industry data used in this study. Section 5 discusses the results obtained for different machine learning models including feature analysis. A discussion of the main outcomes of this study is provided in Section 6.

2 PROBLEM FORMULATION

2.1 Industry background

Booking a major holiday is typically a yearly or bi-yearly activity for travellers, requiring research for destinations, activities and pricing. According to a study from Expedia [12], on average, travellers visit 38 sites up to 45 days prior to booking. The travel sector is characterized by Burke and Ramezani [5] as a domain with the following factors:

- Low heterogeneity: the needs that the items can satisfy are not so diverse.
- High risk: the price of items is comparatively high.
- Low churn: the relevance of items do not change rapidly.
- Explicit interaction style: the user needs to explicitly interact with the system in order to add personal data. Although some implicit preferences can be tracked from web activity and

*Both authors contributed equally to this research.

past history, mainly the information obtained is gathered in an explicit way (e.g. when/where do you want to travel?).
- Unstable preferences: information collected from the past about the user might be no longer trustworthy today.

Researchers have tried to relate touristic behavioural patterns to psychological needs and expectations by 1) defining a characterization of travel personalities and 2) building a computational model based on a proper description of these profiles [27]. Recommender systems are a particular form of information filtering that exploit past behaviours and user similarities. They have become fundamental in e-commerce applications, providing suggestions that adequately reduce large search spaces so that users are directed toward items that best meet their preferences. There are several core techniques that are applied to predict whether an item is in fact useful to the user [4]. With a content-based approach, items are recommended based on attributes of the items chosen by the user in the past [3, 26]. In collaborative filtering techniques, recommendations to each user are based on information provided by similar users, typically without any characterization of the content [19, 24, 25]. More recentely, session-based recommenders have been proposed, where content is selected based on previous activity made by the user on a website or application [17].

2.2 Terminology

In order to clearly define our goal, let us first define some terminology:

- **Hotel Conversion**: a hotel recommendation leads to a conversion when the user books a specific hotel.
- **Hotel Model**: machine learning model trained to predict the conversion probability of individual hotels.
- **Passenger Name Record (PNR)**: digital record that contains information about the passenger data and flight details.
- **Session**: after a traveller completes a flight booking through a reservation system, a session is defined by the context of the flight, the context of the reservation, and a set of five recommended hotels proposed by the recommender system.
- **Session Conversion**: a session leads to a conversion when the user books any of the hotels suggested during the session.
- **Session Model**: machine learning model trained using features related with the session context and hotels, its output is the conversion probability of the session.

The end goal of the recommender system is to increase session conversion. We can estimate the probability of booking of a list of hotels using the session model, and thus we can compare different lists using the session model to determine the one which will maximise the probability of conversion of the session. Note that in this case conversion is defined as a selection or "click" of a hotel on the interface, rather than a booking.

2.3 Hotel recommendations

The content sold through a GDS is diverse, including flight segments, hotel stays, cruises, car rental, and airport-hotel transfers. The core GDS business concerns the delivery of appropriate travel solutions to travel retailers. Therefore, state-of-the-art recommendation engines capable of analysing historical bookings and automatically recommending the appropriate travel solutions need to be designed. Figure 1 shows an outline of the rule-based recommendation system currently in use. After a user books a flight, information related to the trip is sent to the recommender engine.

However, this system does not take into account valuable information such as the context of the request (e.g. where did the booking originate from?), details about the associated flight (e.g. how many days is the user staying in the city?) nor historical recommendations (e.g. are similar users likely to book similar hotels?), which are key assets to fine tune the recommendations.

The problem is novel due to the richness of available data sources (bookings, ratings, passenger information) and the variety of distribution channels: indirect through travel agencies or direct (website, mobile, mailbox). However, it is important to consider that by design, no personally identifiable information (PII) or traveller specific history is used as part of the model, which therefore excludes collaborative-filtering or content-based approaches. The contributions of this work are:

- The combination of data feeds to generate the context of travel, including flights booked by traveller, historical hotels proposed and booked at destination by other travellers, and hotel content information.
- The definition of a 2-stage machine learning recommender tailored for travel context. Two machine learning models are required to build the new recommendation set. The output of the first machine learning algorithm (prediction of the probability of hotel booking) is a key input for the second algorithm, based on the idea of [13].
- The comparison of several machine learning algorithms for modelling the hospitality conversion in the travel industry.
- The design and implementation of a recommendation builder engine which generates the hotel recommendations that maximize the conversion rate of the session. This engine is built based on the analysis of the feature importance of the session model at individual level [29].

3 METHODOLOGY

3.1 Pipeline

Using machine learning and the historical dataset of recommendations, we can train a model which is capable of predicting with high confidence whether a proposed set of recommended hotels leads to a booking.

Once we have fit the model, we can evaluate other combinations of hotels and recommend a list of hotels to the user that maximizes the conversion. Instead of proposing a completely new set of hotels, we decide to modify the existing suggestions given by the existing rule-based system. Our approach, shown in Figure 2, removes one of the initial hotels and introduces an additional one that increases the conversion probability:

We have identified two different ways to select the hotel that is going to be introduced within the set of recommendations:

- We can create and evaluate all possible combinations and choose the one with the highest conversion probability. This means, each time one out of the five hotels from the initial list is removed, and a new one from the pool of hotels is inserted. However, this brute force solution is computationally inefficient and time-consuming (e.g., in Paris this results in

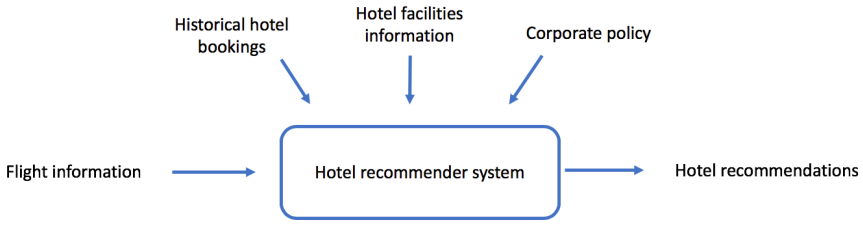

Figure 1: A hotel recommendation system. When a flight booking is completed, the flight details are passed to the hotel recommender engine which selects a set of available hotels for the user based on historical hotel bookings, hotel facilities and a corporate policy check.

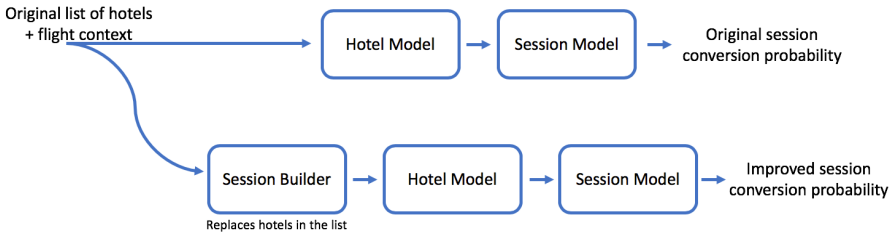

Figure 2: The goal of the system is to improve the probability of conversion. To provide a better set of recommendations, the session builder replaces hotels in the original list.

5*1,653 different combinations for a single swap, the length of the list multiplied by the number of available hotels).

- Alternatively, a hotel from the list of selected hotels can be replaced with an available hotel, based on some criteria. Typically, the criteria might be the price of the hotel room, or the average review score, or a combination of multiple indicators. In this work, the criteria used to optimise the overall list of hotels is determined via feature analysis.

Nevertheless, the last solution presents some challenges that need to be discussed and solved:

(1) How to study the feature importance of complex non-linear models?
(2) How to best interpret the feature importance in an unbalanced dataset?
(3) How many features should be used during the selection process of building an optimal list? Initially, we are facing a multi-objective optimization problem since the choice of a hotel for enhancing the conversion probability might depend on different features. Furthermore, the existence of categorical features makes this optimization even harder. Can we convert it into a univariate optimization problem?

The novelty of this study comes from the use of two related works to address the above points. First, we design a two-stage cascaded machine learning model [13] where the output probabilities of the first model are a new feature of the second one. Second, we interpret the feature importance of the positive instances (i.e. conversions) with a local interpretable model-agnostic (LIME) technique [29]. Thus, we can study the feature importance of particular instances in complex models, allowing the switch from a multi-objective to a univariate optimization problem when one feature is dominant.

3.2 Cascade Generalization

Ensembling techniques consist in combining the decisions of multiple classifiers in order to reduce the test error on unseen data. After studying the bias-variance decomposition of the error in bagging and boosting, Kohavi observed that the reduction of the error is mainly due to reduction in the variance [21]. An issue with boosting is robustness to noise since noisy examples tend to be misclassified and therefore the weight will increase for these examples [2]. A new direction in ensemble methods was proposed by Gama and Brazdil [13] called Cascade Generalization. The basic idea is to use sequentially the set of classifiers (similarly to boosting), where at each step, new attributes are added to the original data. The new attributes are derived from the probability class distribution given by the base classifiers.

There are several advantages of using cascade generalization over other ensemble algorithms:

- The new attributes are continuous since they are probability class distributions.
- Each classifier has access to the original attributes and any new attribute included at lower levels is considered exactly in the same way as any of the original attributes.
- It does not use internal cross validation which affects the computational efficiency of the method.
- The new probabilities can act as a dimensionality reduction technique. The relationship between the independent

features and the target variable are captured by these new attributes.

As will be shown in further sections, this last point is a key aspect of the proposed system, as the probabilities generated by the hotel model can be used to directly select new hotels to include in the recommendation. However, the session model uses aggregated features from the hotel model, and as such an interpretable feature analysis is required to determine how best to select hotels based on their features.

3.3 Interpretability in Machine Learning

Machine learning has grown in popularity in the last decade by producing more reliable, more accurate, and faster results in areas such as speech recognition [16], natural language understanding [8], and image processing [22]. Nevertheless, machine learning models act mostly as black boxes. That is, given an input the system produces an output with little interpretable knowledge on how it achieved that result. This necessity for interpretability comes from an incompleteness in the problem formalisation meaning that, for certain problems, it is not enough to get the solution, but also how it came to that answer [11]. Several studies on the interpretability for machine learning models can be found on the literature [1, 15, 32].

3.4 Local Interpretable Model-Agnostic Explanations (LIME)

In this section, we focus on the work from Ribeiro et al. [29] called Local Interpretable Model-Agnostic Explanations. The Local Interpretable Model-Agnostic Explanations model explains the predictions of any classifier (model-agnostic) in a interpretable and faithful manner by learning an interpretable model locally around the prediction:

- **Interpretable**. In the context of machine learning systems, we define interpretability as the ability to explain or to present in understandable terms to a human [11].
- **Local fidelity**. Global interpretability implies describing the patterns present in the overall model, while local interpretability describes the reasons for a specific decision on a unique sample. For interpreting a specific observation, we assume it is sufficient to understand how it behaves locally.
- **Model-agnostic**. The goal is to provide a set of techniques that can be applied to any classifier or regressor in contrast to other domain-specific techniques [33].

In practice, LIME creates interpretable explanations for an individual sample by fitting a linear model to a set of perturbed variations of the sample and the resulting predictions as output from a complex-model.

3.5 Predictive Models

The selection of which machine learning model to use highly depends on the problem nature, constraints and limitations that are trying to be solved. In this work, algorithms from different families of machine learning were investigated. Specifically, the Naive Bayes Classifier (NB) and Generalised linear Model (GLM) were investigated as linear models, Random Forests (RF), Gradient Boosting Machines (GBMs) were used to evaluate Decision Tree based ensembles and fully connected Neural Networks (NN) were also assessed. Furthermore, the model ensembling technique of Stacking (STK) was also assessed. Stacking comprises of learning a linear model to predict the target variable based on the output probabilities of multiple machine learning algorithms as features.

3.6 Hotel Model

The first step is to train a machine learning model on individual hotels, as shown is Figure 3. The features used for training this model are not exclusively related to hotels, but also with the session and flight context. Evaluating this model, we get the probability that a certain hotel will be booked for a given location. The model is learned by framing the problem as a supervised classification problem, using the conversion (i.e. click) as a label. Note that for the hotel model, the probabilities of conversion are independent of other hotels presented in the session. This leads to several advantages:

- Cold start problem: the model does not penalise items or users that have not been recommended yet, since no hotel identifier or personally identifiable information is used. [31].
- Dimensionality reduction: the output probabilities of the hotel model can be interpreted as a feature that comprises the relationship between the independent variables and the target variable. This is a key concept of the Cascade Generalization technique, thus the output of the hotel model is combined with the features to create the feature vector for the session model, as shown in 4.

Note that the features used as input to the hotel model are discussed in Section 4.

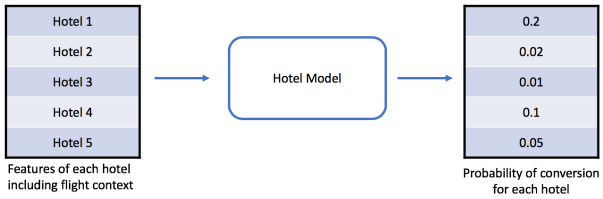

Figure 3: Sketch of the Hotel Model. The machine learning model is trained to predict the probability that each hotel will be booked.

3.7 Session Model

The second machine learning model predicts whether a session leads to a conversion or not, see Figure 4. A session is composed of five different hotels and the aim of the recommender system is to propose a set of hotels that results in the user booking any one of them. Aggregates of the features from the Hotel Model (contextual, passenger, and hotel features) are used, as well as the hotel probabilities obtained from the hotel model. The numerical features related with the hotels are aggregated in different ways (max, min, std and avg of price and probability for example). The features related with the context do not change (e.g. attributes about the session or the flight) as these are identical for each element in the session.

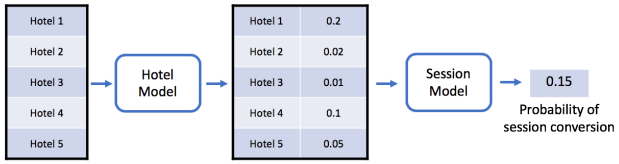

Figure 4: Sketch of the session model pipeline. This machine learning model predicts the probability that a session leads to a conversion, given a list of hotels. This is achieved using cascaded machine learning in which the hotel model predictions are used as features to the session model.

3.8 Session Builder

The Session Model estimates the conversion probability of the session using contextual and content information. Thus, part of the session builder is to create and evaluate new lists of hotels to determine whether these lists will result in higher conversion probability than the original list. Figure 5 shows how this process is performed. First, a reference session with the recommendations, given by an existing rule based system, is scored. For each of the proposed hotels, we estimate the booking probability of each hotel using the Hotel Model. Next, we can calculate the booking probability at session level, using the probabilities of the Hotel Model as an input feature of the Session Model. Then, we aim to improve the conversion probability of the session by removing one of the hotels from the list and introducing a new one. After including the new hotel, if the booking probability of the current session is greater than the probability of the previous session, then this new hotel list is the one that will be proposed to the user.

A rule must be defined to select the hotel to remove and which new hotel to introduce in the recommendation list. Once we have trained the Session Model, we can analyse the feature importance of the variables for the positive cases that were correctly classified (i.e. true positive cases). With the Local Interpretable Model-Agnostic Explanations model [29], we can understand the behaviour of the model for these particular instances. Based on the importance of features from LIME, a heuristic can be defined to replace a hotel from the list in order to improve the session conversion probability.

Note that the LIME analysis is performed only on true positive cases from the training set. In this dataset, the classes are highly imbalanced due to a low conversion rate, as such standard feature analysis techniques may be overly influenced by negative samples, i.e., sessions which did not result in clicks. As LIME is designed to be used on individual decisions, a linear model is fitted and analysed for each true positive. The feature weights for each linear model are then averaged, given a feature importance ranking for all correctly classified converted sessions.

3.9 Evaluation Metrics

As with many conversion problems, the classes are highly imbalanced, and as such the metrics used to assess performance must be carefully chosen.

F-measure (F_β). The generalization of the F_1 metric is given by [7]:

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}$$

β is a parameter that controls a balance between precision P and recall R. When $\beta = 1$, F_1 comes to be equivalent to the harmonic mean of P and R. If $\beta > 1$, F becomes more recall-oriented (by placing more emphasis on false negatives) and if $\beta < 1$, it becomes more precision oriented (by attenuating the influence of false negatives). Common used metrics are the F_2 and $F_{0.5}$ scores.

Area Under the ROC curve. The receiver operating characteristic (ROC) curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold levels. However, this can present an optimistic view of a classifier performance if there is a large skew in the class distribution because the metric takes into account true negatives.

Average Precision (AP). The precision-recall curve is a similar evaluation measure that is based on recall and precision at different threshold levels. An equivalent metric is the Average Precision (AP) which is the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold as the weight:

$$AP = \sum_n (R_n - R_{n-1}) P_n$$

Precision-recall curves are better for highlighting differences between models for unbalanced datasets due to the fact that they evaluate the fraction of true positives among positive instances. In highly imbalanced settings, the AP curve will likely exhibit larger differences and will be more informative than the area under the ROC curve. Note that the relative ranking of the algorithms does not change since a curve dominates in ROC space if and only if it dominates in PR space [10, 30].

4 DATA

4.1 Hotel Recommendation Logs

The dataset in this study consists of 715,952 elements. Out of these recommendations, there are a total of 3,588 clicks, which are considered conversions. Therefore, the dataset is unbalanced since only 0.5% of the instances are session conversions.

Each row contains information regarding the context of the session, the recommended hotel, and whether the recommendation led to a conversion. In particular, the features are the number of recommendations (from 1 to 5), date of the recommendation, country where the booking was made, country where the passenger is traveling, hotel identifier, hotel provider identifier, price of the hotel at time of the recommendation, price currency and whether the recommendation led to a conversion. Additionally, the logs were enriched with supplementary information regarding each hotel including a hotel numerical rating (from 0 to 5), hotel categorical rating and the hotel chain.

4.2 Passenger Name Record

In the travel industry, a Passenger Name Record (PNR) is the basic form of computerized travel record. A PNR is a set of data created when a travel reservation is made. PNRs include the travel itinerary

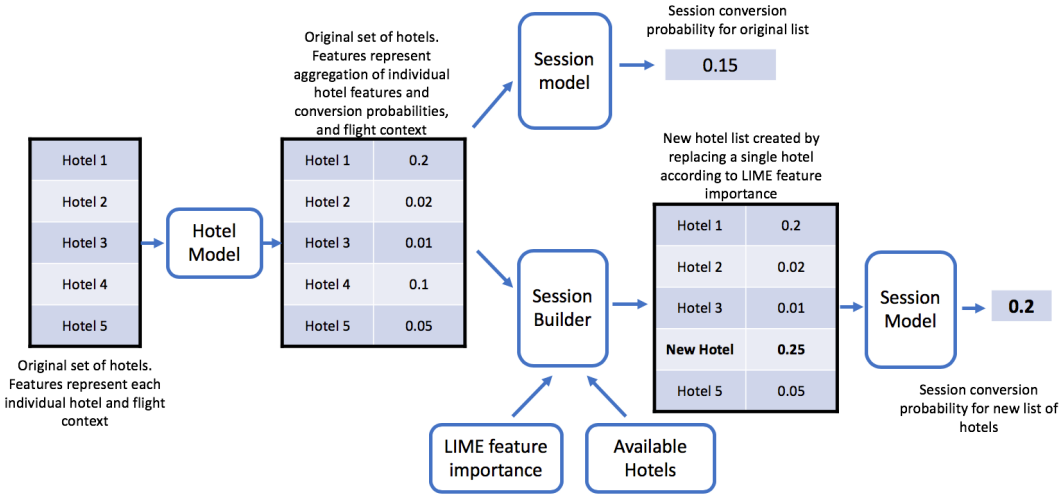

Figure 5: Sketch of the full recommendation pipeline. The session builder is designed to select hotels which will maximise the session conversion, based on the LIME feature importance of the session model.

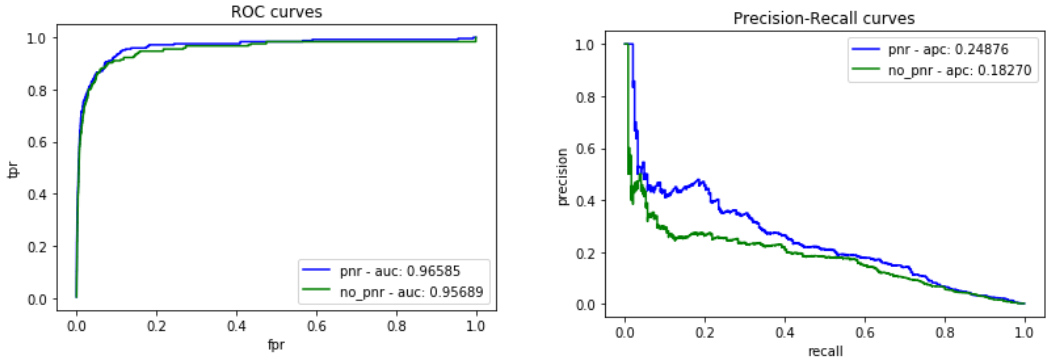

Figure 6: Representation of ROC and AP curves for two Random Forest models predicting individual hotel conversion with and without the PNR data.

information (e.g., flights number, dates) and the passenger information (e.g., name, gender, and somethime passport details). A PNR may also include many other data elements such as payment information (currency, total price, etc), additional ancillary services sold with the ticket (such as extra baggage and hotel reservation) and other airline related information (cabin code, special meal request, etc).

For the purpose of this study, we retrieve and extract features related with the air travel of the traveller. These include the date of PNR creation, airline code, origin city, destination city, date of departure, time of departure, date of arrival, time of arrival, days between the departure and booking date, travel class, number of stops (if any), duration of the flight in minutes (including stops) and the number of days the passenger is staying at the destination.

5 RESULTS

Table 1 shows the results of the experiment comparing different algorithms for the hotel model in terms of AUC, AP, F_1 and $F_{0.5}$ scores. In Figure 6, the ROC and AP curves can be seen in detail. The low AUC value for the GLM model and Naive Bayes Classifier suggest that linear classification techniques do not lead to the best results and more complex models are needed to correctly represent the data. The non-linear techniques have closer results, with the Random Forest obtaining the best values for AP, F_1 and $F_{0.5}$. A Stacked Ensemble using all the previous models is created but it does not improve the previous outcome.

5.1 Contribution of PNR data

The PNR data is an important attribute since it contains rich attributes related to the trip and passenger. However, is this case personally identifiable information is not used in the recommender system, thus the PNR features help to provide context about the

Table 1: Summary of AUC, AP, F_1 and $F_{0.5}$ metrics for the hotel model.

Model	AUC	AP	F1	F0.5
GLM	0.625	0.128	0.247	0.274
NBC	0.819	0.058	0.175	0.159
RF	0.966	**0.249**	**0.320**	**0.334**
GBM	0.953	0.210	0.294	0.288
NN	0.965	0.165	0.245	0.219
STK (all)	0.924	0.182	0.271	0.288
STK (RF + NN)	**0.969**	0.242	0.314	0.284

trip rather than the traveller. Incorporating this data to the models substantially enhanced their performance, as can be observed in Figure 6. Features of the PNR including the number of travellers in the booking and trip duration, among others, contributed to an increase in area under the PR curve from 0.183 to 0.249.

5.2 Session Model

After we have trained the hotel model, we predict individually the probability of conversion of a hotel. Then, we create the sessions based on 5 recommended hotels.

In Table 2 the results are shown. In this case, the best model for both AUC and AP is the Stacked Ensemble composed of a Random Forest, a Generalized Linear Model and a Naïve Bayes Classifier. Although the $F_{0.5}$ score of the GBM model is slightly better than the STK model, the latter clearly outperforms the rest of the metrics.

5.3 Feature Importance

After the Session model has been trained, we analyse its feature importance to study which variables contribute the most to the model using LIME. Concretely, we evaluate the model on the true positive instances from the training dataset, since we want to optimise the conversion.

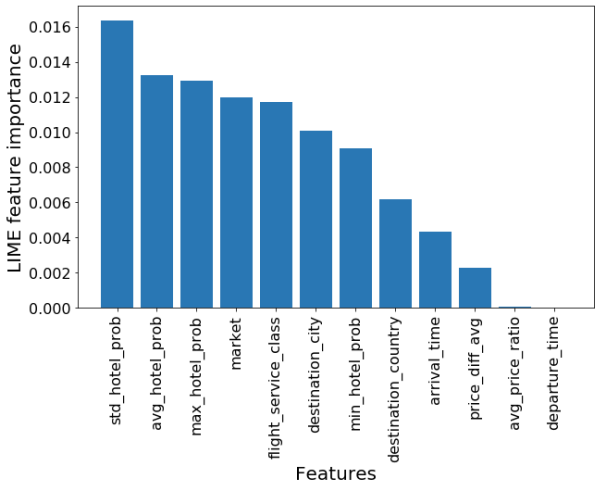

Figure 7: Feature importance of the true positive cases from the Session Model using LIME.

Table 2: Summary of AUC, AP, F_1 and $F_{0.5}$ metrics for the session model.

Model	AUC	AP	F1	F0.5
GLM	0.822	0.395	0.520	0.538
NBC	0.933	0.342	0.467	0.408
RF	0.971	0.446	0.529	0.508
GBM	0.958	0.383	0.531	**0.542**
NN	0.967	0.433	0.483	0.467
STK (RF + GLM + NBC)	**0.972**	**0.453**	**0.539**	0.529

As can be seen in Figure 7, the most important features according to LIME are all derived from the hotel model: the standard deviation, maximum, and average individual hotel conversion probabilities. Some features which are important to the model such as "market" (country where the booking is made from), the flight class of service, the destination city, and arrival and departure times of the flight can not be used to manipulate the results of the session builder, as these are all part of context of the recommendation. Features extracted from prices (the difference between the average price and the minimum, and the ratio of the lowest price to the average price) are also considered important by the LIME model, but rank lower than many hotel conversion probability features.

As the standard deviation of the individual hotel conversions is the most important criteria, the following rule for the session builder is defined: from the original hotel list remove one hotel with the closest conversion probability to the mean conversion probability of the list, and replace it with the hotel with the highest conversion probability from the set of available hotels for a particular city.

5.4 Simulated conversion using Hotel List Builder

Results from the hotel list builder are shown in Table 3 for the two largest cities in the dataset and for the complete dataset. For both cities, we observe a large increase in conversion when using the LIME based session builder. However, a brute force approach to evaluating all possible lists does lead to higher conversion rates, at the cost of a significant increase in processing time. When we consider the complete dataset, we once again observe a large increase in conversion from the baseline for the LIME model. With respect to brute force, we observe that the LIME session builder performs much closer to the brute force builder in terms of conversion. This is attributed to the impact of smaller cities in the complete dataset, and thus less choice in hotels for the builders, resulting in the LIME session builder finding the optimal list. Additionally, on the complete dataset, the processing time of the brute force builder is 2.8 times the duration of the LIME builder, whereas larger gains were observed on the individual cities, where more options for hotels were available.

6 DISCUSSION

In this study, an algorithm was created to improve hotel recommendations based on historical hotel bookings and flight booking attributes. Different machine learning models are used in a cascaded

Table 3: Conversion rates and processing times for two large cities and the complete dataset. The baseline performance is given prior to any optimisation of the hotel lists, the LIME based optimisation is compared to brute force.

	Nice	Barcelona	Complete
Base Conversion	0.0019	0	0.0005
Conversion LIME	0.0207	0.0089	0.0019
Conversion brute	0.0338	0.0125	0.0026
Processing time LIME	23s	23s	4h48m
Processing time brute	314s	496s	13h36m

fashion. First, a model estimates the conversion probability of the individual hotels independently. Note that adding trip context, via PNR based features, resulted in better PR AUC. The output of the first model is then combined with aggregates of the hotels in the list in order to create a feature vector for the session model to estimate the conversion probability that any hotel in the list will be converted. LIME analysis revealed that the hotel model conversion probabilities are the most important features, specifically the standard deviation, mean and maximum individual hotel conversion probabilities in the list. This allows for a simple heuristic to be defined to increase the session conversion probability. In this study, a single change is performed in the list of hotels, however this could be expanded to allow multiple changes.

Variations on this pipeline could also be considered, for instance LIME is used in this study for feature importance ranking in the session builder, however recently a similar methodology was proposed using a mixture regression model referred to as LEMNA [14].

Here, the session builder relies on insights gained from analysis of the feature importance ranking of the session model using LIME over all sessions which lead to a conversion. Thus, the same heuristic is applied to all datapoints in the session builder. However, a key aspect of LIME is that it provides an interpretation of a model for a single datapoint. As such, an evolution of the approach would be to compute the most important features for each recommendation in real time, and to use the information to build an optimal hotel list based on the attributes most likely to lead to conversion.

REFERENCES

[1] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research* 11, Jun (2010), 1803–1831.

[2] Eric Bauer and Ron Kohavi. 1998. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning* 36, 1 (1998), 2.

[3] Yolanda Blanco-Fernandez, Jose J Pazos-Arias, Alberto Gil-Solla, Manuel Ramos-Cabrer, and Martin Lopez-Nores. 2008. Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *IEEE Transactions on Consumer Electronics* 54, 2 (2008).

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (July 2013), 109–132. https://doi.org/10.1016/j.knosys.2013.03.012

[5] Robin Burke and Maryam Ramezani. 2011. Matching recommendation technologies and domains. In *Recommender systems handbook*. Springer, 367–386.

[6] Marcirio Silveira Chaves, Rodrigo Gomes, and Cristiane Pedron. 2012. Analysing reviews in the Web 2.0: Small and medium hotels in Portugal. *Tourism Management* 33, 5 (2012), 1286–1287.

[7] Nancy Chinchor. 1992. MUC-4 Evaluation Metrics. In *Proceedings of the 4th Conference on Message Understanding (MUC4 '92)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 22–29. https://doi.org/10.3115/1072064.1072067

[8] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.

[9] Antónia Correia, Patricia Oom do Valle, and Cláudia Moço. 2007. Why people travel to exotic places. *International Journal of Culture, Tourism and Hospitality Research* 1, 1 (2007), 45–61.

[10] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 233–240.

[11] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. (2017).

[12] Expedia. 2013. Retail and Travel Site Visitation Aligns As Consumers Plan and Book Vacation Packages. https://advertising.expedia.com/about/press-releases/retail-and-travel-site-visitation-aligns-consumers-plan-and-book-vacation-packages

[13] João Gama and Pavel Brazdil. 2000. Cascade generalization. *Machine Learning* 41, 3 (2000), 315–343.

[14] Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. 2018. Lemna: Explaining deep learning based security applications. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 364–379.

[15] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 241–250.

[16] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29, 6 (2012), 82–97.

[17] Dietmar Jannach, Malte Ludewig, and Lukas Lerche. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction* 27, 3-5 (2017), 351–392.

[18] Ingrid Jeacle and Chris Carter. 2011. In TripAdvisor we trust: Rankings, calculative regimes and abstract systems. *Accounting, Organizations and Society* 36, 4 (2011), 293–309.

[19] Michael Kenteris, Damianos Gavalas, and Aristides Mpitziopoulos. 2010. A mobile tourism recommender system. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*. IEEE, 840–845.

[20] Dae-Young Kim, Yeong-Hyeon Hwang, and Daniel R Fesenmaier. 2005. Modeling tourism advertising effectiveness. *Journal of Travel Research* 44, 1 (2005), 42–49.

[21] Ron Kohavi, David H Wolpert, et al. 1996. Bias plus variance decomposition for zero-one loss functions. In *ICML*, Vol. 96. 275–83.

[22] Yann Le Cun, LD Jackel, B Boser, JS Denker, HP Graf, Isabelle Guyon, Don Henderson, RE Howard, and W Hubbard. 1989. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine* 27, 11 (1989), 41–46.

[23] Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. 2012. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 115–122.

[24] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[25] Stanley Loh, Fabiana Lorenzi, Ramiro Saldaña, and Daniel Licthnow. 2003. A tourism recommender system based on collaboration and text analysis. *Information Technology & Tourism* 6, 3 (2003), 157–165.

[26] Raymond J Mooney and Loriene Roy. 2000. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*. ACM, 195–204.

[27] Julia Neidhardt, Leonhard Seyfang, Rainer Schuster, and Hannes Werthner. 2014. A picture-based approach to recommender systems. *Information Technology & Tourism* 15, 1 (sep 2014), 49–69. https://doi.org/10.1007/s40558-014-0017-5

[28] Andreas Papatheodorou. 2001. Why people travel to different places. *Annals of tourism research* 28, 1 (2001), 164–179.

[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1135–1144.

[30] Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one* 10, 3 (2015), e0118432.

[31] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 253–260.

[32] Alfredo Vellido, José David Martín-Guerrero, and Paulo JG Lisboa. 2012. Making machine learning models interpretable.. In *ESANN*, Vol. 12. Citeseer, 163–172.

[33] Peng Zhang, Jiuling Wang, Ali Farhadi, Martial Hebert, and Devi Parikh. 2014. Predicting failures of vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3566–3573.