University of Reading

Department of Computer Science

Predicting Booking Cancellations Using Machine Learning

Samuel Pink

Supervisor: Professor Bryan Lawrence

A report submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Bachelor of Science in *Computer Science*

July 14, 2022

Declaration

I, Samuel Pink, of the Department of Computer Science, University of Reading, confirm that all the sentences, figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

Samuel Pink
July 14, 2022

Abstract

A major problem of revenue management, which exists in almost all booking industries, is the customer's right to cancel the booking before it has been paid; this is equally prevalent in the student accommodation industry. Being able to predict with high accuracy which bookings are going to be cancelled helps to reduce uncertainty and increase revenue. This research was initially started by the aviation industry and has slowly moved into other fields, like the hotel industry. Despite this, there are still a large number of industries that are yet to take advantage of this revenue management strategy. Using a real dataset from the student accommodation industry it has been shown that it is possible to predict with 71 percent recall whether or not an individual booking is going to be cancelled using a machine learning classification model. Results demonstrate that it is not only possible to predict with high accuracy if a booking is going to be cancelled, but it is also possible to identify what specific attributes in the booking cause the cancellation to occur through model explanation. Doing this allows for a more accurate prediction of demand management and revenue forecasting to be made as well as the ability to use this information to prevent bookings from being cancelled.

Keywords: Student Accommodation, Booking cancellations, Machine learning, classification, Azure Auto ML

Report's total word count: 10160

Contents

Chapter 1

Introduction

Machine learning will be used to analyse student accommodation bookings with a focus on the ability to reliably estimate revenue in order to determine when a contract will not be fulfilled. This will be completed utilising data from the property management system Jain et al. (2006).

1.1 Background

A contract made in advanced between the student (customer) and the company, defines the services that will be sold (the room), as well as the start and end date of the tenancy and the fixed price the service will be sold at. In the case of student accommodation, this contract is usually made a few months in advance, in some cases before the student has finalized their plan to attend a specific university. Because of this, the student withholds the right to cancel the contract before it is paid. The inherent uncertainty of the contract means there is no way for the company to definitively measure what the occupancy and sales of a given property will be in advance, since they have to take on the risk of cancellation. This problem is not unique to the student property sector; it is a revenue management issue that was first identified by the aviation industry in 1966 Chiang et al. (2007).

Having the ability to predict with some degree of certainty whether or not a booking is going to be cancelled would allow for more accurate decisions about revenue management to be made and overcome some of the risk created with a contract based booking system.When this is performed with machine learning, the accuracy of the predictions can improve over time as a result of a better understanding of the data, new iterations of modelling, and comparisons of the actual and predicted outcomes. In the case of booking management, a lot of data is stored about each booking made, so the ability of a machine learning algorithm to make accurate predictions on a given outcome is dependent on the data available to it, the amount of data, as well as the accuracy and usability of the data. This means that it should be possible with the correct classification techniques to make accurate predictions on the outcome of the booking.

1.2 Problem Statement

In total for the year 2020 to 2021, 25 percent of all 14363 bookings in the dataset were cancelled, accounting for over 20 million pounds in revenue loss from customers who completed the booking process and then cancelled before the contract was paid. There are a number of different reasons that account for each booking being cancelled that range from impact caused due to Coronavirus and students not receiving their target grades. In most of these cases, the student will then choose to stay at different accommodation resulting in the sale being lost. Producing a model that is able to predict which of these bookings will be cancelled in real time would then allow the business to take actions to try and prevent the student from cancelling their booking. Being able to prevent just 10 percent of users from cancelling would therefore create a potential gain of 2 million in revenue per year.

On a small scale, recognising the characteristics that are common among cancelled bookings would be relatively simple; for example, looking at the percentage of cancelled bookings in the previous year, an approximate estimate of the number of bookings that would be cancelled this year could be estimated. Then, to figure out why these bookings were cancelled, look at what these bookings have in common. In this case, however, no amount of human analysis could possibly account for all of the factors, given that the organisation operates over 70,000 beds worldwide. The fact that a student has the right to cancel a reservation up until a certain time period for a specified penalty means that it is the provider's duty to account for the inherent ambiguity of the contract . The most common method to solve this problem is to simply take the number of bookings cancelled in the previous year and use that number as an average to measure the number of bookings expected to be cancelled in the current year. The problem with this approach is that it only considers one of the many factors, previous year figures, while the company actually keeps hundreds of data points for each booking. Therefore, this problem is going to be attempted to be solved using machine learning. Since the dataset used comes directly from the company in question and includes actual data about students, all personal information has been removed. Instead, students and properties will be referred to solely by their IDs, ensuring that both the company and the students remain anonymous.

1.3 Aims and objectives

By combining all relevant data points from previous years' bookings, a model that can classify a booking into two distinct states, cancelled or not cancelled, is going to be created. It's important to not only build a model that predicts which bookings will be cancelled, but also to understand why bookings are cancelled by using the model to determine which features in the dataset have the greatest impact on whether or not a customer will cancel. This is critical in assisting the company to make business decisions based on the data in order to not only avoid the cancellation but also to understand what actions can be taken to reduce the number of cancellations in the future.

The first step in solving any machine learning problem is obtaining accurate and clean data. In most cases (including this one), this is the most difficult problem to solve because data is rarely stored in a way that makes it easy to read and analyse; in fact, data is frequently stored and then never used. To rectify this, first of all a data warehouse will be created in which to store and analyse the data and then creating a classification model Sessions and Valtorta (n.d.). This data warehouse will exist in the Azure cloud on top of Docker Containers with the code written in Python, with the aim being to create a stream to import new booking data into the data warehouse hourly ensuring that it is stored in a way that makes it easy to use for model training. The aim is to integrate the classification model into the booking management system by creating a daily list of which customers are most likely to cancel their bookings to the managers of their respective properties along with the recommended actions to be taken to prevent the cancellation. Storing the data of the actions taken and whether or not it was successful can then be used to improve future iterations of the model.

1.4 Solution approach

To create a classification model on the booking dataset, the AutoML feature of Azure ML Studio will be used. Auto ML is used because it automates the process of cross validations and feature engineering, which significantly speeds up the model development process. AutoML is able to take advantage of the cloud resources made available through the company by running multiple models at the same time and increasing the speed with which individual models are executed. Bookings will be classified into two distinct groups: cancelled or not cancelled. This is a binary classification because there are only two states in which a booking can be classified. There are two types of classification approaches: supervised and unsupervised learning. This is an example of supervised learning because the expected values are known. Machine learning's aim is to take a set of data with predefined outputs and construct a function that maps the relationship between the inputs and outputs.

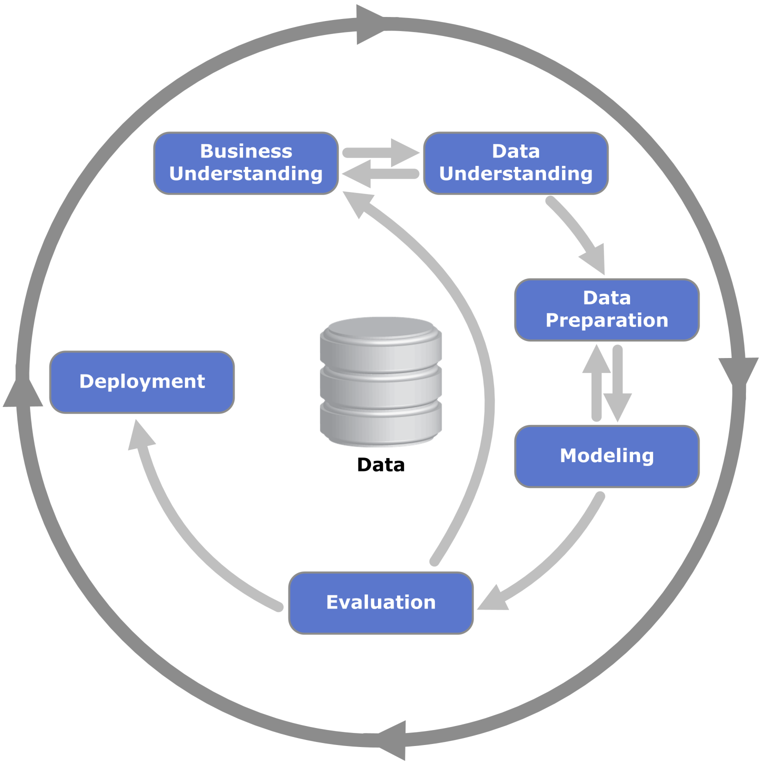


Figure 1.1: The six stages of the CRISP-DM process identified are used to break down the methodology. This diagram depicts the relationship between the various phases of CRISP-DM, demonstrating the sequential design of a data mining project.

The most popular data-mining model is the Cross Industry Standard Method for Data Mining (CRISP-DM), according to current research, due to its numerous advantages, which solved existing data-mining problems Wirth (2000). It works by breaking the problem down into six stages, as shown in Figure 1.1: business understanding, data understanding, data preparation, modelling, evaluation, and deployment.

To determine the best approach to solving the problem, first the company employees were consulted to gain a better understanding of the types of results that would be most effective, as well as the actions that can be taken to have the best chance of preventing a cancellation at the individual booking level. It was discovered that it is critical the results can be presented in a way that employees can understand so that appropriate actions can be taken. Creating a table that ranked bookings by highest probability of cancellation was also discovered to be the best way to display the model's results. Another feature that was found to be important is the ability to retrain the model on new data and adjust the hyper tuning parameters, because the booking dataset is constantly being updated with new entries and it is constantly evolving.

One of the problems that needed to be solved was how to get access to data that is suitable for machine learning, the data preparation stage. This stage consistently requires the most time as seen in previous literature Zhang et al. (2003). To perform machine learning, data needs to be accessible in a suitable format and stored appropriately. The current booking system behind the dataset in this research does not store the data in a way that is suitable for use in machine learning. Building a data warehouse allows for the data to be converted into an appropriate format for machine learning problems. The ability of the data warehouse to update upon new bookings and provide instant access to the data, a feature of the data warehouse using Azure cloud that was used in this project, is an important aspect. The main technology used throughout the modeling process will be Auto ML studio as a framework to encapsulate all of the tools needed to complete a machine learning project, it allows for direct connection to the database for accessing the data through integrated Jupiter Notebooks. Use of Jupiter Notebooks in Python provides an environment for the data cleaning and preparation stages that need to be performed entirely within the azure cloud. The modeling stages are done in the same way by importing the Auto ML Python SDK into the Jupiter Notebook. The Experiment class (azureml.core.experiment.Experiment) in the SDK is used to store and run the models within a logical resource group with the ability to add and remove different modeling iterations from the experiment as well as viewing the results. When a model is configured through AutoMLConfig (azureml.train.automl.automlconfig.AutoMLConfig), it is then added to the experiment and the run is executed in the cloud. Undertaking this process in the cloud helps to solve the General Data Protection Regulation (GDPR) related problems of data access as the data is never stored on the local computer.

The intended outcome of this research is to have the ability to rank bookings by the probability of them being cancelled, providing a platform to integrate the model's results into the company's booking system. Doing this would allow the company to have the ability to input whether or not the prediction was correct. The model would then be able to be retrained based on the outcomes of the predictions, over time allowing for the models accuracy to be increased.

Chapter 2

Literature Review

2.1 Business Perspective

Revenue management is defined as “the application of information systems and pricing strategies to allocate the right capacity to the right customer at the right price at the right time” Kimes and Wirtz (2003). This is essential to a company selling a fixed number of rooms, setting the price too high would result in customers moving to competitors and cause reduced profits as well as a poor customer experience. Setting the price too low however may result in maximum occupancy but will trade this for potential losses in overall profits.

Booking cancellations are one of the main focuses in the revenue management industry as a whole Subramanian et al. (1999). Since a trade-off needs to be made between implementing a ridged cancellation policy that imposes penalties on cancellation, and a policy that has no impact to the customer. Jinhong and Gerstner (2007) shows that implementing a rigid cancellation policy can act as a sales inhibitor and therefore reduce the number of customers. This however further increases the uncertainty as there is little impact to customers cancelling bookings. Talluri and Van Ryzin (2004) finds that in some cases customers looking for the best price will make multiple bookings and then cancel all but one.

Looking at this problem specifically in the student accommodation industry, more focus needs to be directed towards capacity and price, this is unlike the hotel industry which deals more with short term stays. In the student accommodation data used for this research the mean stay is 247 days and once the room is sold it is set at this fixed price for the duration of the academic year.

Demand forecasting is one of five revenue management issues; the others are pricing, auctions, capacity management, and overbooking. The development of a booking cancellation prediction model follows Chiang et al. (2007) recommendation that sales management should use statistical and projection models to best use available data and technologies.

As mentioned by *Revenue management (Book, 2006) [WorldCat.org]* (n.d.) and Weatherford and Kimes (2003), having the ability to accurately forecast demand is key to revenue management in determining how many rooms will be sold and to calculate the correct price at which to sell rooms. This ability to forecast demand then needs to be used in creating the correct pricing model ”in which a perishable and nonrenewable set of resources satisfy stochastic price sensitive demand processes over a finite period of time” Bitran and Caldentey (2003).

Using what was identified by Talluri and Van Ryzin (2004) that “science and technology now make it possible to manage demand on a scale and complexity that would be unthinkable through manual means”; this study aims to use the tools now available in the field of machine learning and the increased number of data points on each customer to approach the prediction of cancellations as a classification problem. This backs up "Demand-based pricing is under-used in many service industries" Kimes and Wirtz (2003) implying that a limited number of industries have properly taken advantage of new technology available in the field of machine learning.

Newell and Marzuki (2018) stated "Amongst the alternative property sectors, student accommodation has recently become an important institutionalised property sector". In the UK alone the student accommodation industry is worth around 60 Billion *UK Student Accommodation Report — United Kingdom — Cushman & Wakefield* (n.d.) with an increase of 14 percent over the last 5 years *UK Student Accommodation Report — United Kingdom — Cushman & Wakefield* (n.d.). Despite this, there is no research in the field of forecasting or analysing demand. With the recent impact of COVID-19 causing a large number of students to have to study from home and in some cases no longer needing student accommodation it is more important than ever to have the ability to accurately forecast the demand.

2.2 Choice of algorithm

Machine Learning algorithms use statistics to find correlations for large quantities of data, if data can be processed in digital form and fed into a machine learning algorithm.Chen (2019). Machine learning algorithms make use of data to increase accuracy and to simulate accurately. Teaching a system can be much simpler than manually programming by showing samples of desired input behaviour in anticipation of the requested answer with all potential inputs.Jordan and Mitchell (2015).

Using machine learning to forecast bookings can be separated into two specific types of problems, regression problems where the aim is to predict a continuous quantitative value like the sum of bookings that will be cancelled in a given year, and classification where each booking can be classified into two groups: cancelled or not cancelled. Most of the research in this field is focused on solving it as a regression problem with Romero Morales and Wang (2010) stating “it is hard to imagine that one can predict whether a booking will be cancelled or not with high accuracy simply by looking at Passenger Name Record (PNR) information”. PNR data does not store the same number of data points as property management system data, it is a file containing information about a passenger (or a travelling group) and their travel plans in the system’s database.

XGBoost is a distributed gradient boosting library that has been optimised for performance, flexibility, and portability Chen and Guestrin (2016). It uses the Gradient Boosting framework to implement machine learning algorithms. XGBoost is a parallel tree boosting algorithm that solves a variety of data science problems quickly and accurately. XGBoost is commonly used by data scientists to produce cutting-edge performance on a variety of machine learning problems. Various interfaces are supported by XGBoost, including the Python interface. The implementation of XGBoost focuses on computational speed, model performance and memory resources.The most critical element in XGBoost’s performance is its scalability in all situations. On a single machine, the device is more than ten times faster than current common solutions, and it scales to billions of examples in distributed or memory-limited environments Chen and

Guestrin (n.d.).

At its most basic level, decision tree analysis is a strategy for classification that can be used to identify and remove important features and patterns from databases to allow for discrimination and modelling predictions. These characteristics, combined with their intuitive interpretation, explain why decision trees have been widely used for over two decades in exploratory data analysis and predictive modelling applications Myles et al. (2004). Due to its efficiency, accuracy, and interpretability, gradient boosting decision trees (GBDT) are a widely used machine learning algorithm. GBDT outperforms the competition in a variety of machine learning tasks. With the advent of big data in recent years, GBDT has encountered new challenges, most notably in the accuracy-efficiency trade-off Ke et al. (n.d.).

2.3 Summary

Current research focuses on looking at prediction cancellations mainly in the hotel and aviation industry, expanding this use of machine learning on revenue management into the student accommodation industry provides more knowledge on how machine learning can be applied to other industries.

Chapter 3

Methodology

3.1 Data warehouse

To obtain the necessary data for solving this problem, the larger issue of developing a reliable data warehouse had to be addressed by creating a dynamic data stream that could take data from an external source and store it in the company data warehouse, allowing new iterations of the model to be run as new data arrived. To facilitate the problem of having a dynamic data import stream, the python object relation mapping library was used since it has direct support for the python data library pandas . Doing this allows data to be read in any supported format and stored in a SQL database with the correct data types. The data needed is stored in an Amazon S3 bucket, so the python S3 library Boto3 was used to read TSV files from blob storage every hour and stored them in the data warehouse. This process was then deployed onto a docker container using the Azure cloud tool called Container Instances that enables docker containers to run as scheduled tasks. Docker containers are standardised units of software that contain all of the dependencies for running a specific program or script in a single executable package which can be run in the cloud or on a local machine.

3.2 CRISP-DM Methodology

As previously mentioned, the CRISP-DM model (Wirth (2000)) is going to be used to break down the problem into 6 specific stages (Figure 1.1).

3.3 Business Understanding

This section asks what the company requires and hopes to gain from the process; in this case, the company requires a better understanding of which customers are most likely to cancel their reservations. This will help in not only preventing cancellations but also in gaining a better understanding of why cancellations occur. Understanding why bookings are being cancelled through the process of feature and model explanation is also important from a machine learning and business perspective, as this will help to understand what the best actions are to prevent cancellations.

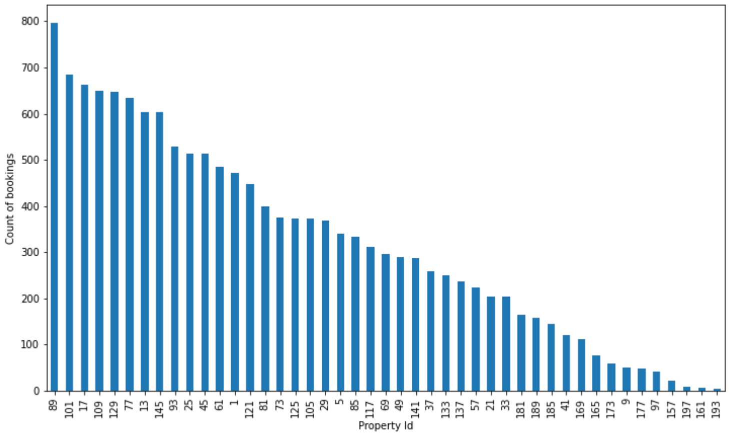


Figure 3.1: Bar graph showing the total number of bookings made in the student accommodation industry for the 2020-2021 time period for each of the properties listed in this dataset. Properties are represented by their property IDs. A large variation in the number of customers per property can be seen, ranging from 796 to 3.

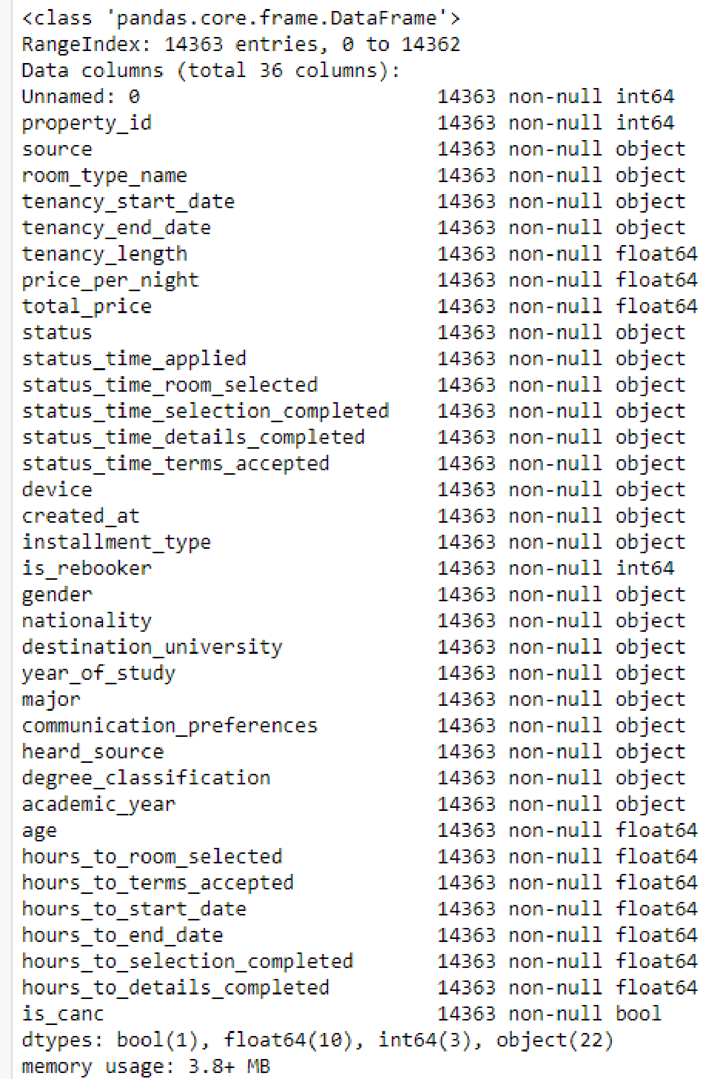
The dataset contains 45 properties in total, with the number of students at each one varying significantly. Because of the large variation, it will be best to understand cancellations at the global level, with one model trained on data from all properties, as shown in Figure 3.1. This helps to avoid model overfitting, as creating an accurate model with a small number of data points on some of the properties would be difficult. The risk here is that some of the discrepancies between the individual properties will go unnoticed by the model.

3.4 Data Understanding

The relevant booking data for this problem is created in an external system, which is the customer-facing website where each booking is taken. Customer information is stored in external website databases when a booking is made through the website. Personal information about the customer, such as name, address, date of birth, and phone number, is included in this data. As the customer progresses through the booking process; they will be directed to a page where they will be asked which university they will be attending and to select the property they wish to rent. The location of the property, the name of the room, the price, and any extras are all stored here. When a student chooses a room, they will be asked about the payment structure they prefer and how they plan to make their deposit. All of the relevant cost data is kept in this folder. In some cases, the fields for data entry in the booking system are not required, implying that the data will not be saved; in these cases, the value of the attribute is replaced with other.

The information stored during these stages of the booking describes all of the intellectual property stored about each individual customer, and it is this information that will be used to forecast the customer's activity and whether or not they will cancel their reservation. In the case of a classification problem, any attribute that influences the target variable is relevant, since the goal of machine learning is to collect as much useful data as possible

In this case, only data from the years 2020 to 2021 will be analysed since the historic data is incomplete. During this booking cycle there are a total of 14,363 bookings.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14363 entries, 0 to 14362
Data columns (total 36 columns):
Unnamed: 0                     14363 non-null int64
property_id                    14363 non-null int64
source                         14363 non-null object
room_type_name                 14363 non-null object
tenancy_start_date             14363 non-null object
tenancy_end_date               14363 non-null object
tenancy_length                 14363 non-null float64
price_per_night                14363 non-null float64
total_price                    14363 non-null float64
status                         14363 non-null object
status_time_applied            14363 non-null object
status_time_room_selected      14363 non-null object
status_time_selection_completed    14363 non-null object
status_time_details_completed  14363 non-null object
status_time_terms_accepted     14363 non-null object
device                         14363 non-null object
created_at                     14363 non-null object
installment_type               14363 non-null object
is_rebooker                    14363 non-null int64
gender                         14363 non-null object
nationality                    14363 non-null object
destination_university         14363 non-null object
year_of_study                  14363 non-null object
major                          14363 non-null object
communication_preferences      14363 non-null object
heard_source                   14363 non-null object
degree_classification          14363 non-null object
academic_year                  14363 non-null object
age                            14363 non-null float64
hours_to_room_selected         14363 non-null float64
hours_to_terms_accepted        14363 non-null float64
hours_to_start_date            14363 non-null float64
hours_to_end_date              14363 non-null float64
hours_to_selection_completed   14363 non-null float64
hours_to_details_completed     14363 non-null float64
is_canc                        14363 non-null bool
dtypes: bool(1), float64(10), int64(3), object(22)
memory usage: 3.8+ MB
```

Figure 3.2: This diagram identifies all of the attributes in the student accommodation booking dataset within the pandas core Data Frame. Non-null indicates that all of the instances of the attributes have values. The data types for each attribute is indicated by the last column.

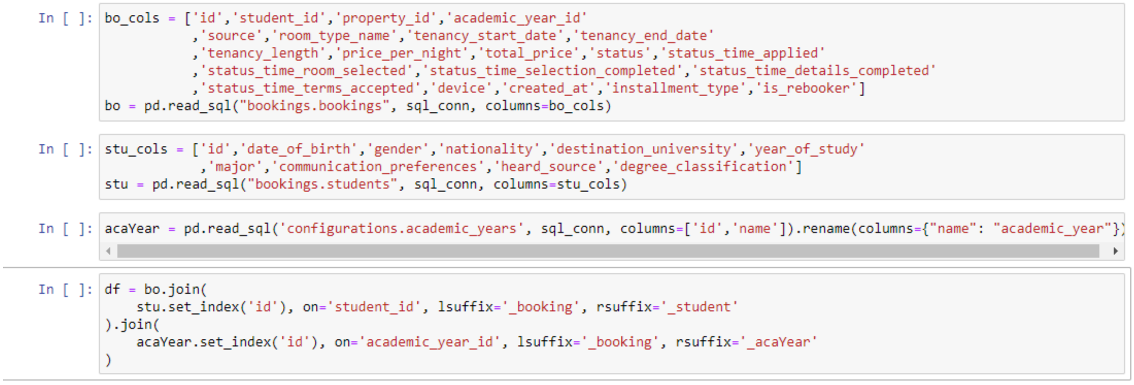
Figure 3.2 shows the features of the booking dataset:

- Property id is the unique identifier for the residence
- Source is the internal system used to create the booking
- Room type name is the category of room selected
- Tenancy start date is the proposed start date of the contract
- Tenancy end date is the proposed end date of the contract
- Tenancy length is the duration in days the contract is valid for
- Price per night is the daily rate at which the room will be sold
- Total price is the total price for the contracted time

- Status is the current status of the booking
- Status time applied is the time the booking process was started
- Status time room selected is the time the customer selected their room
- Status time selection completed is the time the customer finalised the selected process
- Status time details completed is the time all personal details are entered
- Status time terms accepted is the time the agreement is completed
- Device is the type of the device the booking was made on
- Created at is the time the process was started
- Installment type is the payment schedule
- Is rebooker defines if the same customer has applied before
- Date of birth of the customer
- Gender of the customer
- Nationality of the customer
- Destination university is the university the customer expects to go to
- Year of study is the academic year the customer is in
- Major is the degree type of the student
- Communication preference is the customer selected method of communication
- Heard source is where the customer discovered the booking
- Degree classification is the degree type of the customer
- Academic year

3.5 Data Preparation

The dataset selected to be used for this research is a combination of 3 different tables from the external booking system, these include the booking table, student table and academic year ID table. All of these tables have been combined from the data warehouse to create the Data Frame shown in Figure 3.2. During the stage of feature selection many of the attributes have been removed from these tables as they did not provide any value and did not contain good data quality. Using a dataset with too many columns has the potential to result in over fitting and noise in the model; it is possible that the dataset selected can be reduced to a smaller number of features but the best dataset available has been selected for completing this research.

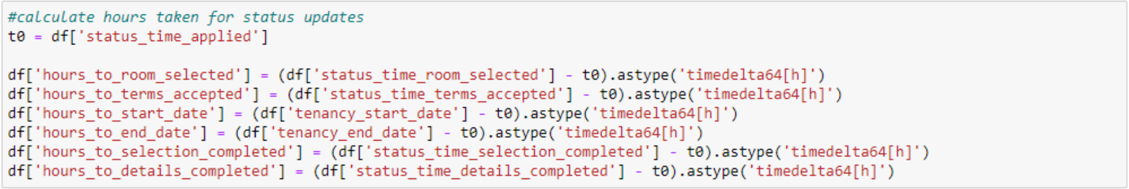


```
In [ ]: bo_cols = ['id','student_id','property_id','academic_year_id'
                   ,'source','room_type_name','tenancy_start_date','tenancy_end_date'
                   ,'tenancy_length','price_per_night','total_price','status','status_time_applied'
                   ,'status_time_room_selected','status_time_selection_completed','status_time_details_completed'
                   ,'status_time_terms_accepted','device','created_at','installment_type','is_rebooker']
        bo = pd.read_sql("bookings.bookings", sql_conn, columns=bo_cols)

In [ ]: stu_cols = ['id','date_of_birth','gender','nationality','destination_university','year_of_study'
                    ,'major','communication_preferences','heard_source','degree_classification']
        stu = pd.read_sql("bookings.students", sql_conn, columns=stu_cols)

In [ ]: acaYear = pd.read_sql('configurations.academic_years', sql_conn, columns=['id','name']).rename(columns={"name": "academic_year"})

In [ ]: df = bo.join(
            stu.set_index('id'), on='student_id', lsuffix='_booking', rsuffix='_student'
        ).join(
            acaYear.set_index('id'), on='academic_year_id', lsuffix='_booking', rsuffix='_acaYear'
        )
```

Figure 3.3: This shows the stage of data preparation from a section of Jupiter Notebook, used for student accommodation data cleaning, where 3 SQL tables were joined into one data frame using their respective ID's

With all of the necessary data stored in the data warehouse, the relevant tables were combined and imported into a Jupiter Notebook to perform the data cleaning stages (Figure 3.3). The aim is to include only valid bookings that made it the whole way through the booking process. Initially, any booking in the dataset that didn't have a total price or with a total price less than one was removed as this meant the booking was not stored in the system correctly and may have been used for testing. Then, any booking that did not get to the terms accepted stage was removed since this could not be treated as a cancellation or a booking as the customer process was not finished.

To account for the missing values in the categorical columns, the pandas fill na method was used to replace the null values with 'Other' to prevent errors from occurring during the modeling stages when trying to handle null values and to make it clear in the results when a missing value had been used.



```
#calculate hours taken for status updates
t0 = df['status_time_applied']

df['hours_to_room_selected'] = (df['status_time_room_selected'] - t0).astype('timedelta64[h]')
df['hours_to_terms_accepted'] = (df['status_time_terms_accepted'] - t0).astype('timedelta64[h]')
df['hours_to_start_date'] = (df['tenancy_start_date'] - t0).astype('timedelta64[h]')
df['hours_to_end_date'] = (df['tenancy_end_date'] - t0).astype('timedelta64[h]')
df['hours_to_selection_completed'] = (df['status_time_selection_completed'] - t0).astype('timedelta64[h]')
df['hours_to_details_completed'] = (df['status_time_details_completed'] - t0).astype('timedelta64[h]')
```

Figure 3.4: This shows a section of the Jupiter Notebook used for feature engineering the student accommodation dataset by measuring the time taken to complete each stage of the booking process in the number of hours from the status time applied.

Figure 3.4 shows the feature engineering applied by using the status time applied column to act as the first point where the booking process was started. The attributes were created to store how long the customer took within each stage of the booking process as this describes the customer's activity throughout the booking process, for example if a customer took only 1 hour to complete the booking process they may be less likely to cancel than one who took 2 days.

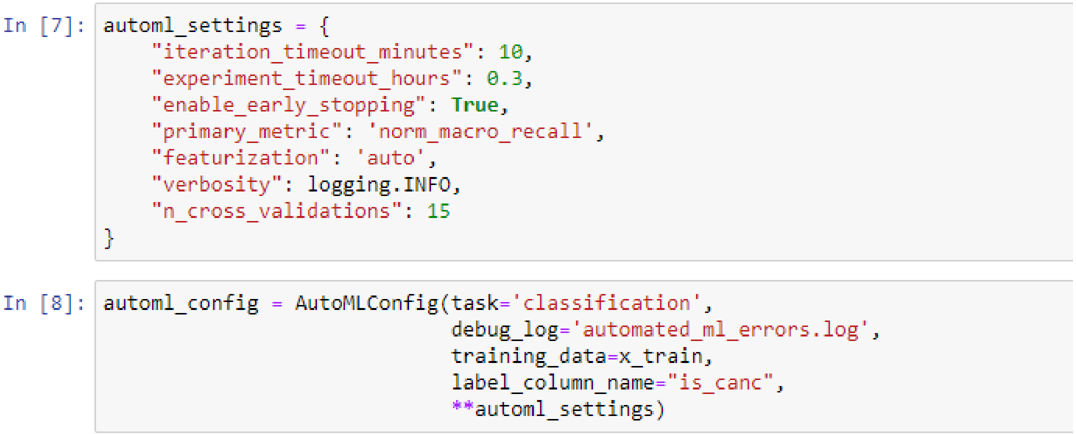
Using the featurization auto setting allows Auto ML to automatically detect the column types and apply the optimal data preprocess techniques; for categorical data the One Hot encoding technique is applied to create a new variable for each stage of the categorical attribute represented as binary. The process of featurization is able to detect columns with high cardinality. High cardinality means an attribute contains a large number of unique values; this was true for columns destination university, major and nationality. This is an expected result as the dataset contains students from a large variety of universities and nationalities. When a feature with high cardinality is detected by Auto ML it is automatically dropped from the dataset.

To handle missing numeric values, imputation is applied by the automatic featurization in Auto ML by replacing any missing values with the average of the column. Imputing missing values working best in columns with only a small proportion of missing values, this is why all columns with a significant proportion of missing values have been dropped before this stage. Data scaling is a pre-processing stage, real-valued input and output variables may be normalised or standardised to achieve data scaling. Normalization was used to scale each input variable independently to the range 0-1. This estimator scales and translates each feature separately, resulting in a maximum absolute value of 1.0 for each feature in the training set. It does not shift or centre the data, so there is no loss of sparsity Singh (2019), *sklearn.preprocessing.MaxAbsScaler — scikit-learn 0.24.1 documentation* (n.d.)

3.6 Modeling

Auto ML Studio was used to evaluate and compare multiple different algorithms. Auto ML studio is a cloud environment used to train, deploy, automate, manage and track Auto ML models. It can be used for multiple different types of machine learning like supervised or unsupervised and deep learning. It gives the ability to write code in Python, R and its own no-code environment. The cloud Jupiter notebook features were used for the data cleaning and preparation stages as well as testing models. To evaluate and compare multiple different algorithms, Auto ML was used through its Python SDK. Doing this meant a large variation of regression algorithms were able to be tested.

A train test split of 20 percent test size was used with the sklearn model selection library. 80 percent training data was used to ensure there was a sufficient amount of data to ensure good model accuracy.



```
In [7]: automl_settings = {
            "iteration_timeout_minutes": 10,
            "experiment_timeout_hours": 0.3,
            "enable_early_stopping": True,
            "primary_metric": 'norm_macro_recall',
            "featurization": 'auto',
            "verbosity": logging.INFO,
            "n_cross_validations": 15
        }

In [8]: automl_config = AutoMLConfig(task='classification',
                                     debug_log='automated_ml_errors.log',
                                     training_data=x_train,
                                     label_column_name="is_canc",
                                     **automl_settings)
```

Figure 3.5: This illustrates the final AutoML configuration settings used for running the classification algorithm on the student accommodation dataset.

Running the algorithm is setup using the AutoMLConfig class in the Azure ML python SDK, the object contains all of the parameters used for configuring the experiment run. For a classification algorithm, the task attribute is set to classification along with passing the training data and the target column label. Setting the number of cross validations to 15 means AutoML will run 15 different classifications on the training data outputting the results of each iteration. This output is then ranked by the selected primary metric which has been selected as norm macro recall, AutoML optimises models selected based on the primary metric.

After the configuration is setup it can be passed to the Experiment.submit class. In this case, the local run configuration was used so the cloud resources were not consumed running many iterations of the classification model. As each model is generated its results are outputted to the screen.

3.7 Evaluation

The model results will be evaluated based on the number of bookings accurately predicted as going to cancel, as this is the metric that best satisfies the problem specification. A successful model will be one that can classify over 50 percent of cancelled bookings accuracy. This relatively low target accuracy is used because this is the first research completed in this specific field and therefore being able to accurately predict over 50 percent of cancelled bookings shows that it is in fact possible to use a classification algorithm in predicting if a booking is going to be cancelled. Evaluating the CRISP-DM methodology needs to be concerned with understanding how each of the individual stages contributed to solving the final business objective. In most machine learning projects data understanding and data preparation have the most overall impact with a relatively small amount of time being taken in the modeling stage Polyzotis et al. (2018). Evaluating how changes in the data being inputted into the model affects its ability to accurately predict cancellations.

3.8 Deployment

The deployment of the model will be handled by AutoML by registering the final model into the Azure cloud allowing it to then be accessed through the Azure API and integrated into any web supported system. The deployment of the model into the cloud is not within the scope of this project.

Chapter 4

Results

This project uses a machine learning model with sklearn and Auto ML to classify bookings from a dataset in the student accommodation industry as either cancelled or not cancelled. The goal of this study is to develop a prediction for each booking as to whether it will be cancelled, allowing for data-driven decisions in the hopes of reducing cancellations.

4.1 Exploratory data analysis

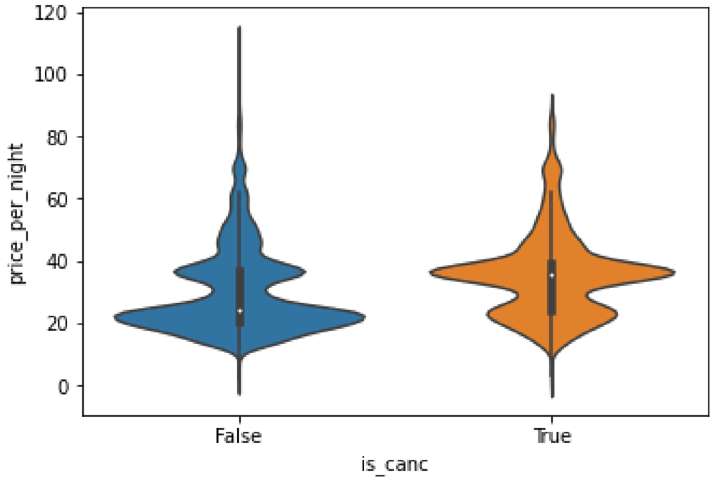


Figure 4.1: A violin plot illustrating the distribution of price per night for student accommodation bookings, comparing cancelled and non cancelled bookings. False (blue) represents non-cancelled and True (orange) represents cancelled bookings.

Figure 4.1 compares the distribution of price per night when looking at bookings that cancelled to ones that did not. The mean price per night for bookings that cancelled is £35 per night compared to £29 per night for not cancelled bookings. This difference in price distribution when comparing cancelled to non cancelled bookings suggests price per night is an important feature for classification.

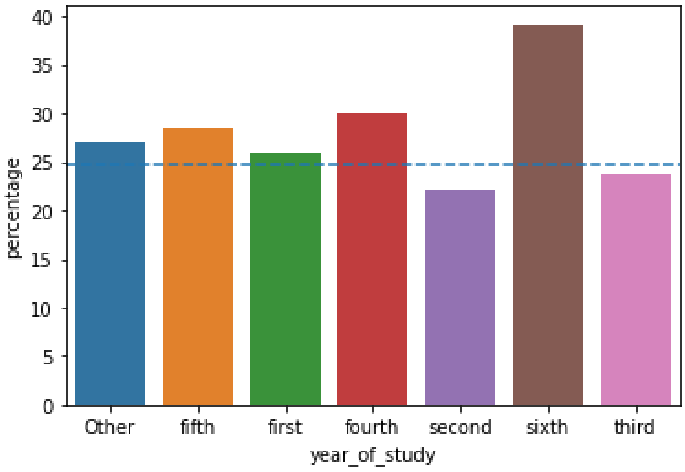


Figure 4.2: A bar graph showing the relative frequencies of the unique values as a percentage of booking cancellations in the student accommodation industry. Data is plotted by degree year of study with the blue horizontal line representing the mean cancellations.

Figure 4.2 shows percentage of booking cancellations grouped by academic year of study. This shows sixth years as having 40 percent cancellations, this may be due to the fact they are more likely to decide to live in private accommodation but it is unlikely this accounts for them being almost twice as likely to cancel than the average. The Other category is also an outlier here with a 27 percent chance of cancelling, more research would be needed to understand exactly why customers who do not enter a year of study are more likely to cancel but it is clear from this figure that year of study will be a good indicator of cancellations.

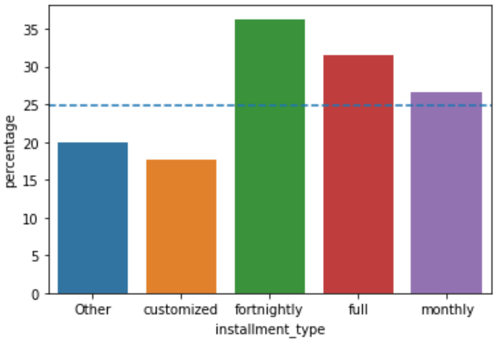


Figure 4.3: A bar graph illustrating the relative frequencies of the unique values as a percentage of booking cancellations for payment instalment type in the student accommodation industry booking dataset. The blue horizontal line indicates the mean cancellation. Full payment plans represent the student playing all at once, monthly and fortnightly payment plans indicate the students paying at regular time intervals and customised payments allow the student to create their own plan.

Figure 4.3 shows the weighted average of cancellations looking at the selected instalment type. Instalment type is the payment plan the student selected, we can see that students who selected the fortnightly plan are 10 percent more likely to cancel than the average, and students who opted for the customized instalment plan are 7 percent less likely to cancel.

This may be a good predictor of cancellations.

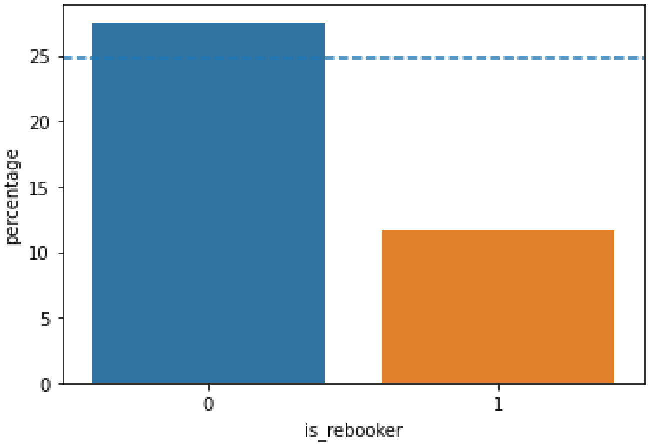
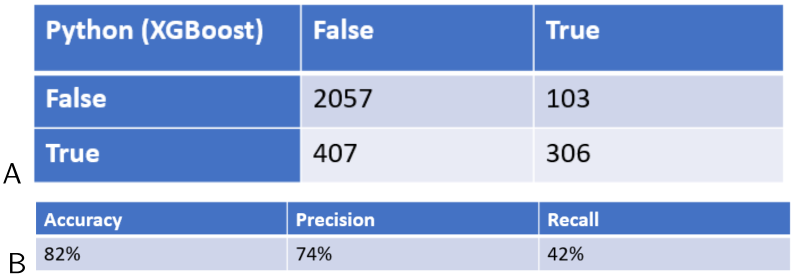


Figure 4.4: A bar graph illustrating the relative frequencies of the unique values as a percentage of booking cancellations for a rebooker, someone who has booked before. 0 (blue) represents a individual who has not booked before, 1(orange) represents someone who has.

Figure 4.4 shows the comparison of cancellations between students who are rebookers (have booked before) and those who are not. We can see that students who have not booked previously are 16 percent more likely to cancel their bookings. This is another good predictor of cancellations.

4.2 Original Model - XGBoost



A

Python (XGBoost)	False	True
False	2057	103
True	407	306

B

Accuracy	Precision	Recall
82%	74%	42%

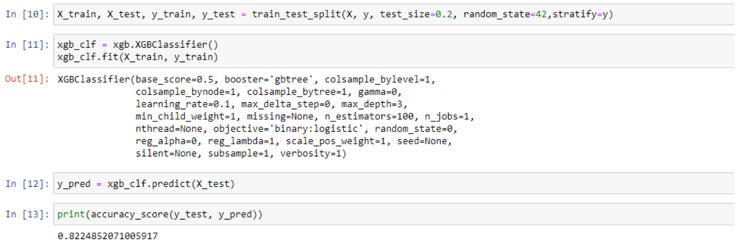
Figure 4.5: **These tables illustrate the original XGBoosts methods results.**

A = The confusion matrix produced from the student accommodation booking data. False False: a non cancelled booking correctly predicted, False True: booking predicted as not cancelled but actually cancelled. True True: predicted as cancelled and actually cancelled and True False: predicted as cancelled but not actually cancelled.

B = A table illustrating the accuracy, precision and recall of the XGBoost method as a percentage.

The confusion matrix in Figure 4.5 is a table that displays the performance of a classification model on a set of test data where the true values are known. The 4 sections that make up a confusion matrix are, true positives (TP) these are bookings that the model predicted as cancelled and are actually cancelled. True negatives (TN) these are bookings predicted correctly as not cancelled. False positives (FP) are bookings that the model predicted as cancelled but did not and False negatives (FN) these are bookings the model predicted as not cancelled but did cancel. TP is the most important because this is where the model has predicted a booking as going to cancel and has actually cancelled which is the objective of this research.

The accuracy of a classification model is (TP+TN)/total. This is the number of bookings accurately classified as either cancelled or not cancelled. In this case it is more valuable to classify TP than TN since cancelled bookings are the focus of this research. Recall is TP / (TP + FN), this is the proportion of actual positive values predicted correctly.



```
In [10]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42,stratify=y)

In [11]: xgb_clf = xgb.XGBClassifier()
         xgb_clf.fit(X_train, y_train)

Out[11]: XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                       colsample_bynode=1, colsample_bytree=1, gamma=0,
                       learning_rate=0.1, max_delta_step=0, max_depth=3,
                       min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
                       nthread=None, objective='binary:logistic', random_state=0,
                       reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
                       silent=None, subsample=1, verbosity=1)

In [12]: y_pred = xgb_clf.predict(X_test)

In [13]: print(accuracy_score(y_test, y_pred))

         0.8224852071005917
```

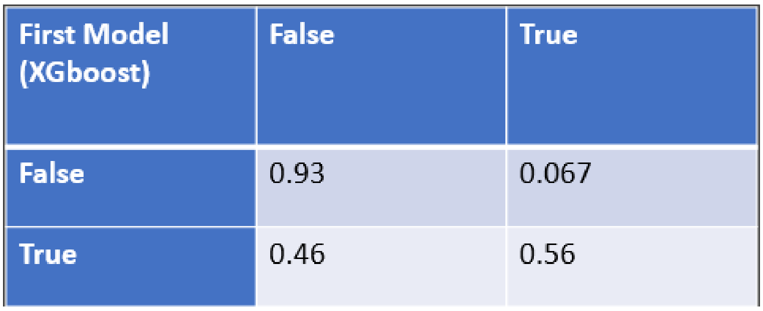
Figure 4.6: The original code used for running the XGBoost model. A test size of 20 percent was used and the XGBoost classifier was created and then fit to the training data. Prediction shows 82 percent accuracy overall on the test data.

The original methodology included using Python to run XGBoost, Figure 4.6 shows a snippet of the code used to train the model with a train test split set to 20 percent. The model was created on the test data and the accuracy score was calculated. The model was run using the default hyper parameters, in sklearn. Initially, the accuracy score of 82 percent appeared to solve the problem specification. However, the confusion matrix shown in Figure 4.6 identified that there is a large number of false positive predictions compared with the number of true positive predictions, showing that the models recall is low compared to the accuracy. As recall identifies the number of true positives, correct cancellations, a low recall score is undesirable.

After running this model and identifying poor recall, an alternative model and Hyper-parameter optimization needed to be identified to produce increase recall. Using Auto ML, different algorithms were tested to enable the best algorithm to be identified.

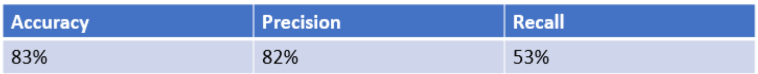
4.3 Auto ML

Since Auto ML is automated, it allows for more effective testing and evaluation of different models and hyper parameter combinations. This addresses the previous issue of not knowing which model and hyper parameter combination to use in order to solve this problem.



First Model (XGboost)	False	True
False	0.93	0.067
True	0.46	0.56

A



Accuracy	Precision	Recall
83%	82%	53%

B

Figure 4.7: **These tables illustrate the first Auto ML model results.**

A= The confusion matrix produced from the student accommodation booking data presenting as normalized. False False: a non cancelled booking correctly predicted, False True: booking predicted as not cancelled but actually cancelled. True True: predicted as cancelled and actually cancelled and True False: predicted as cancelled but not actually cancelled.

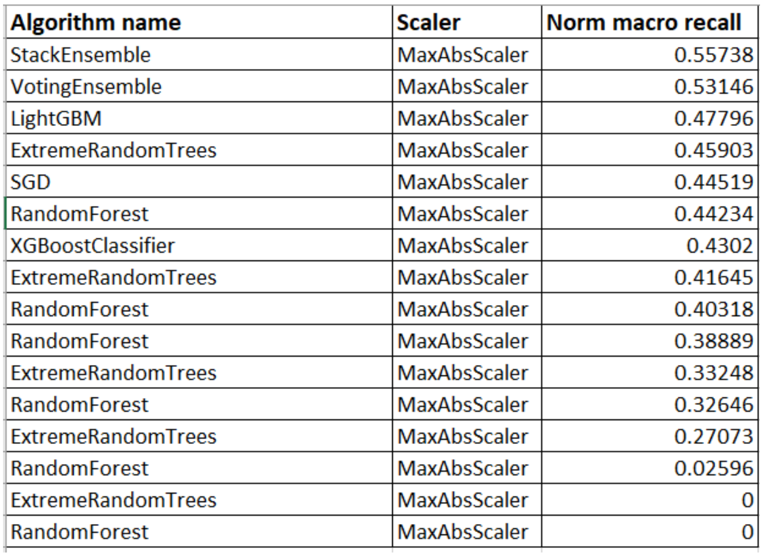
B= A table illustrating the accuracy, precision and recall of the first Auto ML method.

Figure 4.7 shows 0.93TF 0.56TP gives an overall accuracy of 0.83, this is inline with the results of running XGBoost manually outside of Auto ML. Figure 4.7 also shows only 53 percent of true positive values are identified correctly in the first run of Auto ML. This is only slightly better than the results of the original XGBoost model, with a high accuracy and low recall the original problem statement has not been solved.

4.4 Auto ML- Stack Ensemble

Stack Ensemble learns how to merge predictions using a meta-learning algorithm from two or more base machine learning algorithms. The benefit of the stack is the fact that a variety of high performance models incorporate the capabilities to predict that any single model in the ensemble is superior to one classification. This is made up from 2 or more base models and a meta model that combines the predictions of the base models.

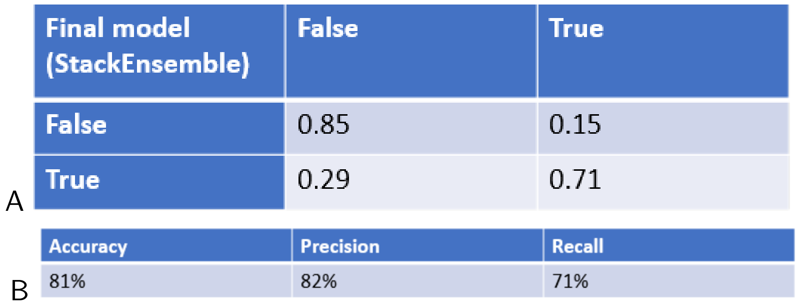
Knowing that recall should be the primary metric to target and not accuracy, using Auto ML, the target metric was changed to recall. Figure 3.5 shows the settings applied with the new primary metric being normalized macro recall. Setting the primary metric means Auto ML can optimise the model selection based on recall.



Algorithm name	Scaler	Norm macro recall
StackEnsemble	MaxAbsScaler	0.55738
VotingEnsemble	MaxAbsScaler	0.53146
LightGBM	MaxAbsScaler	0.47796
ExtremeRandomTrees	MaxAbsScaler	0.45903
SGD	MaxAbsScaler	0.44519
RandomForest	MaxAbsScaler	0.44234
XGBoostClassifier	MaxAbsScaler	0.4302
ExtremeRandomTrees	MaxAbsScaler	0.41645
RandomForest	MaxAbsScaler	0.40318
RandomForest	MaxAbsScaler	0.38889
ExtremeRandomTrees	MaxAbsScaler	0.33248
RandomForest	MaxAbsScaler	0.32646
ExtremeRandomTrees	MaxAbsScaler	0.27073
RandomForest	MaxAbsScaler	0.02596
ExtremeRandomTrees	MaxAbsScaler	0
RandomForest	MaxAbsScaler	0

Figure 4.8: A table outlining the different algorithms used for the final run of Auto ML and the scaler used on the student accommodation booking data. Models were ranked by their recall (norm macro recall).

During this run Auto ML was used to evaluate and compare multiple different algorithms knowing that the algorithm with the best recall needed to be used, Auto ML was run again keeping all other values the same, and only changing the primary metric. Auto ML performs 15 different cross-validations on the data set, ranking them by the target attribute, recall (Figure 4.8). The same dataset and train test split were used, meaning the model evaluates and ranks the results differently.



A

Final model (StackEnsemble)	False	True
False	0.85	0.15
True	0.29	0.71

B

Accuracy	Precision	Recall
81%	82%	71%

Figure 4.9: **These tables illustrate the final Auto ML, Stack Ensemble model results.**

A = The confusion matrix produced from the student accommodation booking data presented as normalized. False False: a non cancelled booking correctly predicted, False True: booking predicted as not cancelled but actually cancelled. True True: predicted as cancelled and actually cancelled and True False: predicted as cancelled but not actually cancelled.

B = A table illustrating the accuracy, precision and recall of the final Auto ML, Stack Ensemble method.

The confusion matrix in Figure 4.9 shows a significant improvement in recall when compared to the confusion matrix in Figure 4.6. This outlines the benefit of using the Stack Ensemble algorithm compared to XGBoost.

In comparison to Figure 4.6, the final confusion matrix has a lower overall accuracy score but a higher recall score. The recall is a representation of the true predicted cancellations, a higher recall score is more desirable. The final model had a recall score of 71 percent (Figure 4.9) which is 18 percent higher than the original model.

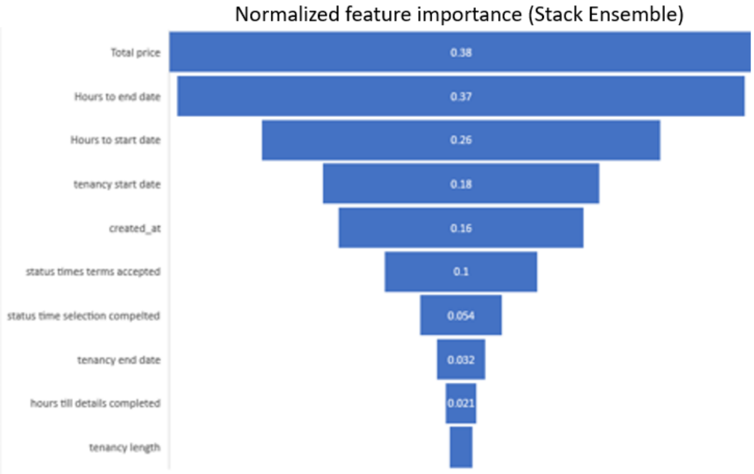
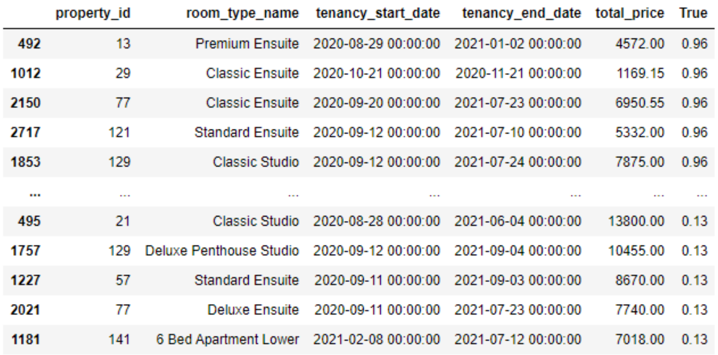


Figure 4.10: A bar graph showing the feature importance of the final Auto ML model, ordered from high to low.

Based on the results from the feature importance graph in Figure 4.10, the main dependent feature on cancellations was total price, this was consistent with the feature exploration graph in Figure 4.1 which indicated price per night to be the most important feature. However, the rest of the features indicated in the exploratory data analysis were not present in the feature importance graph in Figure 4.10. The majority of the features in the feature importance graph were based on numerical data, such as hours to start or end date.



	property_id	room_type_name	tenancy_start_date	tenancy_end_date	total_price	True
492	13	Premium Ensuite	2020-08-29 00:00:00	2021-01-02 00:00:00	4572.00	0.96
1012	29	Classic Ensuite	2020-10-21 00:00:00	2020-11-21 00:00:00	1169.15	0.96
2150	77	Classic Ensuite	2020-09-20 00:00:00	2021-07-23 00:00:00	6950.55	0.96
2717	121	Standard Ensuite	2020-09-12 00:00:00	2021-07-10 00:00:00	5332.00	0.96
1853	129	Classic Studio	2020-09-12 00:00:00	2021-07-24 00:00:00	7875.00	0.96
...	...	...	...	...	...	...
495	21	Classic Studio	2020-08-28 00:00:00	2021-06-04 00:00:00	13800.00	0.13
1757	129	Deluxe Penthouse Studio	2020-09-12 00:00:00	2021-09-04 00:00:00	10455.00	0.13
1227	57	Standard Ensuite	2020-09-11 00:00:00	2021-09-03 00:00:00	8670.00	0.13
2021	77	Deluxe Ensuite	2020-09-11 00:00:00	2021-07-23 00:00:00	7740.00	0.13
1181	141	6 Bed Apartment Lower	2021-02-08 00:00:00	2021-07-12 00:00:00	7018.00	0.13

Figure 4.11: A table of prediction probabilities, ranking each of the bookings made by their chance of cancellation, using the Stack Ensemble model on the student accommodation data. Property ID represents the unique property. True represents the chance of the booking being cancelled.

All of the bookings in the test data are ranked between 0 and 1, a booking over 0.5 gets classified as cancelled. The prediction probability was calculated using the Stack Ensemble model and is able to rank bookings made by the chance of them being cancelled.

Chapter 5

Discussion and Analysis

5.1 Findings

The final recall of the model in terms of predicting if a booking made is going to be cancelled is 71 percent, this is similar to what was found by Antonio et al. and shows that machine learning can be used to solve the problem of predicting booking cancellations in the student accommodation industry. Other papers in revenue management look at results in terms of overall accuracy Antonio et al. (2017) of the model not the precision.True True to True False ratio is also important when trying to understand which bookings will not be cancelled as well as which will be. The importance of this depends on the specific research but in this case there is little value in predicting if a booking will not be cancelled.

The results from the exploratory data analysis show that there are significant differences in the dataset between cancelled and non cancelled bookings. These findings can be used to identify the reasons behind cancellations. For example, sixth year students with high price per night bookings who have not booked before appear to be significantly more likely to cancel based on this analysis. Identifying the factors that increase the chance of cancellation can allow for data-driven decisions to be made based on this information to attempt to prevent cancellations. Increasing communication between the accommodation company and students with a high chance of cancellation could potentially prevent cancellations. These findings show that it is possible to accurately train a classification model to predict cancellations.

It was expected that the features with significant differences between cancelled and non cancelled bookings to have a high ranking on the feature importance. However, only price per night out of the features identified in the exploratory data analysis were included in the feature importance. This could be due to the feature engineering performed by Auto ML. Although the majority of the features discovered through exploratory data analysis were categorical, machine learning algorithms work best with numerical data, potentially explaining the differences in the features identified. Future work should focus on feature engineering to integrate the categorical variables more effectively into the model, allowing for the potential identification of other features having a more significant impact than price.

The dataset used in this project is from the time period 2020-2021. There is a chance that the amount of cancellations and the reasons behind these could have been affected by the Coronavirus pandemic, potentially influencing the results of this model. In some cases students have been advised to stay at home and not live in halls during this time period, meaning that the booking would have been cancelled. This could have produced a dataset which is not fully representative of a normal academic year. However, the ratio of cancellations in this dataset was in line with previous research Antonio et al. (2017), suggesting that the pandemic may not have had an affect at all. There was no available data from before the Coronavirus pandemic that could be used in this project. In order to be sure that the dataset was not affected by the Coronavirus pandemic, these findings should be compared to future years. However, there are some valuable insights that can be gained from this project regardless of Coronavirus as it is likely that the pandemic will still have a significant impact on the student accommodation industry in the next academic year.

5.2 Applications of findings

Being able to analyse the 2020-2021 booking data has allowed for the accurate prediction of cancellations as well as identifying the reasons behind them. This can be useful in determining business strategies to implement, aiming to reduce the impact of student accommodation cancellations. The application of these findings are particularly important in the student accommodation industry due to the uncertainty of the booking contract. Student accommodation companies maximise there occupancy based on the number of bookings but do not take into account the amount of cancellations. Having the ability to be able to predict the cancellations with a high degree of certainty will allow for a more accurate forecasting of occupancy, subsequently reducing revenue lost.

Given the fact that in the year 2020/2021 a single company in the student accommodation industry lost £20 million in revenue from customers who completed the booking process but then cancelled, there is a lot of valuable insights to be gained from this area of study. This is just one specific example of how machine learning used correctly can be used to solve uncertainty in business situations. Based on the findings of this study, effective application of this approach in other industries with sufficient data may be beneficial. Applying these tools allow for data driven decision making to be applied into industries that are yet to adapt and take advantage of new technologies that are available. The methodology used shows how the skills and time required for machine learning to be correctly applied have reduced with the development of tools like Auto ML studio that abstract away alot of the complexity in projects similar to this one. The applications of this project are relevant in a variety of industries and will allow for business predictions to be made.

5.3 Methodology alterations

The classification model was implemented using the XGBoost Python library in the original process. According to previous studies, this approach is the most efficient Antonio et al. (2017). When looking at accuracy, XGBoost performs best overall, however, it was not accurately able to predict true positive values. Because of this, an alternative method was used to produce more accurate results.

From a machine learning point of view, using Auto ML is beneficial to be able to test multiple different algorithms with cross-validation and select the one with the best results in a more efficient manner. This is also beneficial from the business perspective as it provides an explanation behind the results, allowing for insights into the models accuracy. Using Auto ML allowed for Stack Ensemble to be identified as the best algorithm to be used in predicting booking cancellations on this dataset which would not have been identified otherwise. This has produced significantly better results in comparison to the original algorithm used, XGBoost. In comparison to using the Python libraries to create the models manually, using Auto ML allowed for the testing of a larger variety of algorithms to gain more accurate results. Auto ML has built in functionality that provides an explanation of the model results which allows for a better understanding of the model output that can subsequently can be used to retrain the model, further improving its accuracy.

As with all machine learning problems, the model is most accurate when trained over a number of iterations, this usually takes a long time in writing the code and preparing the model. Using Auto ML, the iterations of the model can be trained more efficiently and understand the results better. This was particularly beneficial in this project as it allowed for the realisation that targeting the precision was a more accurate method. Originally, the accuracy was targeted which is the total number of correct predictions, both true and false. However, due to the weighting of not cancelled bookings in the test data, a high accuracy was achieved without predicting a significant number of true positive bookings.

As well as altering the classification algorithm, the primary metric was also changed. When looking at the confusion matrix, the number of false positives and true positives were roughly the same, meaning the model had not accurately predicted any of the bookings. Instead, the model accurately predicted bookings that were not going to be cancelled. These findings produced no valuable insights as the standard assumption is that bookings are not going to be cancelled.To produce a more beneficial model, the primary metric was changed to target recall instead. Recall is a more useful insight as it identifies the bookings that are going to be cancelled. Using Auto ML allowed for a straightforward method of changing the target metric. This subsequently allowed for the model to be re run with recall as the target metric. The results of this model were beneficial by allowing for actual cancellations to be predicted.

5.4 Future work

The dataset includes 75 percent of bookings as not cancelled, with only 25 percent being cancelled. This ratio of data produces bias towards the not cancelled bookings which is not ideal when using the data to predict cancellations. If the data was changed to contain a 50:50 split, the model could potentially be more accurate. Having a dataset which is more heavily weighted as cancelled would allow the model to be less biased towards non cancelled bookings. Future research should experiment with different sizes of data-sets to identify at which point the most accurate model would be achieved.

Due to insufficient amount of data for some properties, the model in this project used all properties to predict cancellations. It is likely that creating a model that predicts cancellations on a per property basis would be more accurate. This is reinforced by previous research in revenue management which uses a per property level for their models. A per property model would likely be more accurate as each property has different attributes and different potential reasons for cancellations. For example, the properties in this dataset were in a variety of countries, at different universities and the price varied heavily depending on location. Although the findings of this model were still useful, future research should focus on using more data and making the model property specific.

The amount of time taken for a booking to be made seems to be a significant indicator of cancellations as shown by the feature importance. The impact of the hours to start date on the cancellation process was not identified until analysing the results of this machine learning model. With this knowledge, additional analysis based on exploring the relationship between hours to start date and cancellations could be performed. This would allow for further insights into the specific behaviour of customers who cancel there bookings and ways to prevent this.

Having access to previous years data would allow for more in-depth insights into the year on year trends of booking cancellations and adding this data to the model would also increase the accuracy. As well as analysing the year on year trends, incorporating more data into the model would allow for a better accuracy on predicting cancellations. Future work should expand on adding more of the booking data into the data warehouse.

The next steps for this project include implementing the model into the PMS system to be able to highlight customers on a daily basis who are most likely to cancel their bookings through a ranking system. Using the models feature explanation, the reason why this customer has been selected as most likely to cancel their booking should be able to be identified and with the provided recommended set of messages to send the customer in an attempt to prevent the cancellation from occurring.

With a clean dataset in place that encapsulates the customers data, it may be possible to extend the predictive capabilities to predicting what new properties will be most profitable in the future to assist in company decisions of which properties to buy and which to sell. It would be desirable to apply this classification methodology to other areas of the company.

Chapter 6

Conclusion

6.1 Conclusions

The final model accuracy gives a score of 71 percent true positive bookings, this is consistent with the results of Antonio et al. (2017) when looking at recall and shows that it is in fact possible to predict booking cancellations with relatively high accuracy using PMS data in the context of student accommodation. Proving machine learning can be applied using Azure Auto ML studio to solve regression problems using already existing data sets if care is taken in the data preparation stages, this is the benefit of using the CRISP-DM methodology to break down the problem into a series of well defined stages to solve a specific example of a generic problem.

Chapter 7

Reflection

Throughout this project I have learnt how to implement my knowledge of machine learning into a real world example by using a classification model to solve a business problem of predicting booking cancellations. Doing so has given me a greater understanding of how to judge a model's results, as I used to focus solely on model accuracy, assuming that a model with a higher accuracy score was better. I now know that it is important to gain a deeper understand of why a specific accuracy score is given by looking at the results from the confusion matrix.

I learnt to use a new tool, Auto ML studio, which I believe helped to improve my ability to create and implement machine learning predictions through providing infrastructure that allowed me to easily evaluate and compare many different algorithms.

I found that I was able to solve my original objective by predicting with 71 percent recall if a booking is going to be cancelled, this is significant as previously there was no research looking at bookings in the student accommodation industry. I think the application of machine learning into industries that are yet to take advantage of it can provide a lot of value since there are only a few industries that have been properly able to take advantage of these new tools available in computer science. One of the reasons for this is the complexity involved in the many stages that make up a machine learning project, I believe by taking advantage of Auto ML studio the complexity of solving this problem is reduced.

References

Antonio, N., Almeida, A. d. and Nunes, L. (2017), 'Predicting hotel booking cancellations to decrease uncertainty and increase revenue', *Tourism & Management Studies* **13**(2), 25–39.

Bitran, G. and Caldentey, R. (2003), 'An Overview of Pricing Models for Revenue Management'.
URL: *http://pubsonline.informs.org.https://doi.org/10.1287/msom.5.3.203.16031http://www.informs.org*

Chen, L.-P. (2019), 'Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar: Foundations of machine learning, second edition', *Statistical Papers* **60**(5), 1793–1795.
URL: *https://doi.org/10.1007/s00362-019-01124-9*

Chen, T. and Guestrin, C. (2016), XGBoost: A scalable tree boosting system, *in* 'Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', Vol. 13-17-August-2016, Association for Computing Machinery, pp. 785–794.
URL: *https://github.com/dmlc/xgboost*

Chen, T. and Guestrin, C. (n.d.), 'XGBoost: A Scalable Tree Boosting System'.
URL: *http://dx.doi.org/10.1145/2939672.2939785*

Chiang, W. C., Chen, J. C. and Xu, X. (2007), 'An overview of research on revenue management: current issues and future research', *International Journal of Revenue Management* **1**(1), 97–128.

Jain, K., Sharma, V. and Mehta, S. J. (2006), Intellectual Property Management System: An Organizational Perspective, Technical report.

Jinhong, X. and Gerstner, E. (2007), 'Service escape: Profiting from customer cancellations', *Marketing Science* **26**(1), 18–30.
URL: *https://pubsonline.informs.org/doi/abs/10.1287/mksc.1060.0220*

Jordan, M. I. and Mitchell, T. M. (2015), 'Machine learning: Trends, perspectives, and prospects'.
URL: *http://1.usa.gov/1eNy7qR.*

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y. (n.d.), LightGBM: A Highly Efficient Gradient Boosting Decision Tree, Technical report.
URL: *https://github.com/Microsoft/LightGBM.*

Kimes, S. E. and Wirtz, J. (2003), 'Has Revenue Management become Acceptable?', *Journal of Service Research* **6**(2), 125–135.
URL: *http://journals.sagepub.com/doi/10.1177/1094670503257038*

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A. and Brown, S. D. (2004), 'An introduction to decision tree modeling'.

Newell, G. and Marzuki, M. J. (2018), 'The emergence of student accommodation as an institutionalised property sector', *Journal of Property Investment and Finance* **36**(6), 523–538.

Polyzotis, N., Roy, S., Whang, S. E. and Zinkevich, M. (2018), 'Data lifecycle challenges in production machine learning: A survey', *SIGMOD Record* **47**(2), 17–28.
URL: *https://dl.acm.org/doi/10.1145/3299887.3299891*

Revenue management (Book, 2006) [WorldCat.org] (n.d.).
URL: *https://www.worldcat.org/title/revenue-management/oclc/137238292*

Romero Morales, D. and Wang, J. (2010), 'Forecasting cancellation rates for services booking revenue management using data mining', *European Journal of Operational Research* **202**(2), 554–562.

Sessions, V. and Valtorta, M. (n.d.), THE EFFECTS OF DATA QUALITY ON MACHINE LEARNING ALGORITHMS, Technical report.

Singh, P. (2019), MLlib: Machine Learning Library, *in* 'Learn PySpark', Apress, pp. 85–115.
URL: *https://link.springer.com/chapter/10.1007/978-1-4842-4961-1*$_5$

sklearn.preprocessing.MaxAbsScaler — scikit-learn 0.24.1 documentation (n.d.).
URL: *https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MaxAbsScaler.html*

Subramanian, J., Stidham, S. and Lautenbacher, C. J. (1999), 'Airline yield management with overbooking, cancellations, and no-shows', *Transportation Science* **33**(2), 147–167.
URL: *https://pubsonline.informs.org/doi/abs/10.1287/trsc.33.2.147*

Talluri, K. T. and Van Ryzin, G. J. (2004), *The Theory and Practice of Revenue Management*, Vol. 68 of *International Series in Operations Research & Management Science*, Springer US, Boston, MA.
URL: *http://link.springer.com/10.1007/b139000*

UK Student Accommodation Report — United Kingdom — Cushman & Wakefield (n.d.).
URL: *https://www.cushmanwakefield.com/en/united-kingdom/insights/uk-student-accommodation-report*

Weatherford, L. R. and Kimes, S. E. (2003), 'A comparison of forecasting methods for hotel revenue management', *International Journal of Forecasting* **19**(3), 401–415.

Wirth, R. (2000), 'CRISP-DM : Towards a Standard Process Model for Data Mining', *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining* (24959), 29–39.

Zhang, S., Zhang, C. and Yang, Q. (2003), 'Data preparation for data mining', *Applied Artificial Intelligence* **17**(5-6), 375–381.
URL: *https://www.tandfonline.com/action/journalInformation?journalCode=uaai20*

Appendix A

Appendix

A.1 Auto ML

```
# In[1]:


import pandas as pd
from azureml.core import Dataset
from datetime import datetime
from dateutil.relativedelta import relativedelta

from azureml.core.workspace import Workspace
from azureml.train.automl import AutoMLConfig
from azureml.core.experiment import Experiment

from sklearn.model_selection import train_test_split

import logging


# In[2]:


ws = Workspace.from_config()


# In[3]:


df = pd.read_csv(’290321.csv’)


# In[4]:


df = df.drop(columns=[’status’,’Unnamed: 0’])


# In[5]:


df.info()


```

```
# In[6]:


x_train, x_test = train_test_split(df, test_size=0.2, random_state=223)
print(x_train['is_canc'].value_counts()/len(x_train))
print(x_test['is_canc'].value_counts()/len(x_test))


# In[7]:


automl_settings = {
    "iteration_timeout_minutes": 10,
    "experiment_timeout_hours": 0.3,
    "enable_early_stopping": True,
    "primary_metric": 'norm_macro_recall',
    "featurization": 'auto',
    "verbosity": logging.INFO,
    "n_cross_validations": 15
}


# In[8]:


automl_config = AutoMLConfig(task='classification',
                             debug_log='automated_ml_errors.log',
                             training_data=x_train,
                             label_column_name="is_canc",
                             **automl_settings)


# In[ ]:


experiment = Experiment(ws, "canc-090421")
local_run = experiment.submit(automl_config, show_output=True)


# In[ ]:


from azureml.widgets import RunDetails
RunDetails(local_run).show()


# In[ ]:


best_run, fitted_model = local_run.get_output()
print(best_run)
print(fitted_model)


# In[ ]:


test = x_test.drop(columns=['is_canc'])
best_run, fitted_model = local_run.get_output()
class_prob = fitted_model.predict_proba(test)

```

```

# In[ ]:


class_prob


# In[ ]:


#pd.merge(class_prob, test, left_index=True, right_index=True)
pd.concat([class_prob, test], axis=1)


# In[ ]:


canc_prob = test.join(class_prob).sort_values(by=True, ascending=False).
                                        dropna(subset=[True])


# In[ ]:


canc_prob[True].value_counts(bins=2)


# In[ ]:


canc_prob[[’property_id’, ’room_type_name’,’tenancy_start_date’,’
                                        tenancy_end_date’,’total_price’,True]
                                        ]


# In[ ]:


canc_prob.info()


# In[ ]:
```

A.2 Data Cleaning

```

import pyodbc
import pandas as pd
from sqlalchemy import create_engine
import sqlalchemy
import sys
pd.set_option(’display.max_columns’, None)
pd.set_option(’display.max_rows’, 400)
import matplotlib.pyplot as plt
import seaborn as sns
import seaborn as sn
from datetime import date, timedelta
# Use seaborn style defaults and set the default figure size
```

```
sns.set(rc={'figure.figsize':(11, 4)})


# In[ ]:


username = '****'
password = ****
server = '****'
database = '****'


# In[ ]:


URL = "mssql+pyodbc://" + username + ":" + password + "@" + server + "/" + database + "?driver=ODBC+Driver+17+for+SQL+Server"

sql_conn = create_engine(URL,fast_executemany=True).connect()


# In[ ]:


bo_cols = ['id','student_id','property_id','academic_year_id'
          ,'source','room_type_name','tenancy_start_date','tenancy_end_date'
          ,'tenancy_length','price_per_night','total_price','status','status_time_applied'
          ,'status_time_room_selected','status_time_selection_completed','status_time_details_completed'
          ,'status_time_terms_accepted','device','created_at','installment_type','is_rebooker']
bo = pd.read_sql("bookings.bookings", sql_conn, columns=bo_cols)


# In[ ]:


stu_cols = ['id','date_of_birth','gender','nationality','destination_university','year_of_study'
           ,'major','communication_preferences','heard_source','degree_classification']
stu = pd.read_sql("bookings.students", sql_conn, columns=stu_cols)


# In[ ]:


acaYear = pd.read_sql('configurations.academic_years', sql_conn, columns=['id','name']).rename(columns={"name":"academic_year"})


# In[ ]:


df = bo.join(
    stu.set_index('id'), on='student_id', lsuffix='_booking', rsuffix='_student'
).join(
    acaYear.set_index('id'), on='academic_year_id', lsuffix='_booking', rsuffix='_acaYear'
)


# In[ ]:


df = df.rename(columns={"name": "property_name"})


# In[ ]:


to_dt = ['tenancy_start_date','tenancy_end_date','status_time_applied','status_time_room_selected'
         ,'status_time_selection_completed','status_time_details_completed','status_time_terms_accepted'
         ,'date_of_birth']

df[to_dt] = df[to_dt].apply(pd.to_datetime, format='%Y-%m-%d',errors='coerce')


# In[ ]:


# add age from D.O.B
today = date.today()
df['age'] = df['date_of_birth'].apply(lambda x: today.year - x.year - ((today.month, today.day) <(x.month, x.day)))


# In[ ]:


#calculate hours taken for status updates
t0 = df['status_time_applied']

df['hours_to_room_selected'] = (df['status_time_room_selected'] - t0).astype('timedelta64[h]')
df['hours_to_terms_accepted'] = (df['status_time_terms_accepted'] - t0).astype('timedelta64[h]')
df['hours_to_start_date'] = (df['tenancy_start_date'] - t0).astype('timedelta64[h]')
df['hours_to_end_date'] = (df['tenancy_end_date'] - t0).astype('timedelta64[h]')
df['hours_to_selection_completed'] = (df['status_time_selection_completed'] - t0).astype('timedelta64[h]')
df['hours_to_details_completed'] = (df['status_time_details_completed'] - t0).astype('timedelta64[h]')


# In[ ]:


#remove any booking with invalid price
```

```
df = df[df['total_price'].notna()]
df = df.drop(df[df['total_price'] < 1].index)


# In[ ]:


#remove any booking with invalid age
df = df[df['age'].notna()]
df = df.drop(df[df['age'] < 10].index)


# In[ ]:


#remove any booknig that did not reach terms accepted
df = df[df['status_time_terms_accepted'].notna()]


# In[ ]:


#remove any booking made before it started
df = df.drop(df[df['hours_to_selection_completed'] < 0].index)
df = df.drop(df[df['hours_to_terms_accepted'] < 0].index)
df = df.drop(df[df['hours_to_start_date'] < 0].index)


# In[ ]:


df.installment_type = df.installment_type.fillna('Other')
df.gender = df.gender.fillna('Other')
df.nationality = df.nationality.fillna('Other')
df.destination_university = df.destination_university.fillna('Other')
df.year_of_study = df.year_of_study.fillna('Other')
df.major = df.major.fillna('Other')
df.heard_source = df.heard_source.fillna('Other')
df.degree_classification = df.degree_classification.fillna('Other')


# In[ ]:


df['is_canc'] = df['status'].str.contains('cancelled')


# In[ ]:


df = df.drop(columns=['id','student_id','academic_year_id', 'date_of_birth'])


# In[ ]:


df.to_csv('290321.csv')
```

A.3 XGboost Python

```python


import pandas as pd

from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, f1_score, recall_score
from sklearn import metrics
import numpy as np

import matplotlib.pyplot as plt

import xgboost as xgb

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 400)


# In[2]:


#importing bookings dataset
bo = pd.read_csv('290321.csv')


# In[3]:


drop = ['tenancy_start_date', 'tenancy_end_date', 'status', 'status_time_applied'
        , 'status_time_room_selected', 'status_time_selection_completed', 'status_time_details_completed'
        , 'status_time_terms_accepted', 'created_at']

bo = bo.drop(columns=drop)


# In[4]:


num_vars = ['property_id','tenancy_length','price_per_night','age'
            ,'hours_to_room_selected','hours_to_terms_accepted','hours_to_start_date'
            ,'hours_to_end_date','hours_to_selection_completed','hours_to_details_completed']

cat_vars = ['source','room_type_name','device','installment_type','is_rebooker','gender'
            ,'nationality','destination_university','year_of_study','major'
            ,'communication_preferences','heard_source','degree_classification'
            ,'academic_year']


```

```
# In[5]:


df = pd.get_dummies(bo,columns=cat_vars)


# In[6]:


imr = SimpleImputer(strategy=’mean’)
df[num_vars] = imr.fit_transform(df[num_vars])


# In[7]:


scaler = StandardScaler()

df[num_vars] = scaler.fit_transform(df[num_vars])
df = pd.DataFrame(df)


# In[8]:


regex = re.compile(r"\[|\]|<", re.IGNORECASE)
df.columns = [regex.sub("_", col) if any(x in str(col) for x in set((’[’, ’]’, ’<’))) else col for col in df.columns.values]


# In[9]:


X= df.drop(columns=’is_canc’)
y = df[’is_canc’]


# In[10]:


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42,stratify=y)


# In[11]:


xgb_clf = xgb.XGBClassifier()
xgb_clf.fit(X_train, y_train)


# In[12]:


y_pred = xgb_clf.predict(X_test)


# In[13]:


```

```
print(accuracy_score(y_test, y_pred))


# In[14]:


ft_imp = pd.Series(xgb_clf.feature_importances_,index=X_train.columns).sort_values(ascending=False)
ft_imp[0:15].plot.barh()


# In[26]:


ft_imp[0:11]


# In[33]:


confusion_matrix(y_test, y_pred)


# In[32]:


C.astype(’float’) / C.sum(axis=1)[:, np.newaxis]


# In[16]:


metrics.precision_score(y_test, y_pred)


# In[35]:


recall_score(y_test, y_pred)


# In[17]:


f1_score(y_test, y_pred, average=’weighted’)


# In[18]:


metrics.plot_roc_curve(y_test, y_pred)
plt.show()
```