5 Combining Infrastructure Sensor and Tourism Market Data in a Smart City Project – a case study

Andreas Komninos, Mark D Dunlop and John N Wilson

Tourism is an important industry in many countries, making up for a significant percentage of a country's GDP, and often amounting to the major source of income for local communities. According to one definition, Smart Tourism is "tourism supported by integrated efforts at a destination to collect and aggregate/harness data derived from physical infrastructure, social connections, government/organizational sources and human bodies/minds in combination with the use of advanced technologies to transform that data into on-site experiences and business value-propositions with a clear focus on efficiency, sustainability and experience enrichment" [Gretzel et al. 2015a].

One key aspect of supporting smart-tourism is prediction of visitor numbers to allow more accurate and efficient service support, the focus of this case study. The challenge of the project described here was to understand the problem of forecasting tourism levels in a city destination using multiple data sources. Our case study was based on Glasgow, Scotland. Glasgow is the largest city in Scotland with about 1.6 million inhabitants in the Greater Glasgow area and a strong diverse tourism pull. Visitors come for cultural events (it has several museums & theatres, is home to Scottish Opera, Scottish Ballet & National Theatre of Scotland and has a vibrant music scene from small venues through to stadium concerts), architecture, conference hosting, and as a gateway venue to the Scottish Highlands and golf courses. The city of Glasgow is also home to three Universities and three professional football clubs that add considerable pulls to the city. [Scottish Enterprise and Glasgow City Marketing Bureau 2014].

To understand visitor numbers and report to the government on the Glasgow tourism industry, The Glasgow City Marketing Bureau (GCMB), relies heavily on annual visitor statistics from the UK's Office for National Statistics and Great Britain Tourism Survey, plus other visitor surveys. However, access to this data has a significant lag time. We worked with GCMB to investigate if multiple diverse sources of data could predict tourist numbers and provide advance indicators to hotels, business and other tourism stakeholders. While multiple indicators of tourism levels

Author's Draft

have been used previously, e.g. extrapolating arrival data (e.g. [Chu 2004]), we decided to focus on the level of demand for hotel occupancy and focus on attempting to predict daily figures that would best support hotel planning using a mix including city sensor, flight search, venue and advanced hotel booking data. This is a challenging metric as there are many reasons for visiting Glasgow that result in hotel usage (e.g. business/conference visits, independent tourism, and attending large cultural events) and previous work had focussed on more general week-by-week or month-by-month forecasting (e.g. [Yang et al. 2015; Gunter and Önder 2016]). As well as being a higher level of granularity this added data challenges of aligning with calendars (e.g. varying on day of week and moveable events such as Easter) and major seasonal variation (e.g. University terms).

Previous research, e.g. [Li et al. 2017; Bangwayo-Skeete and Skeete 2015], had focussed on building models of web search activity. We wanted to focus on a mix of more focussed data. We were able to select from a wide variety of data sources available to the city council. However, to bring this data together in a usable form, we needed to begin by understanding the nature of the datasets: the nature and volume of data; reporting frequency; and machine readability. While all the data was available digitally, much was not easily processed as it came in the form of spreadsheets that were prepared for human consumption rather than automatic processing (e.g. inconsistent columns and use of font styling). These low-level processing tasks are common to much of heterogeneous data analytics. Finally, once able to establish a suitable pipeline of pre-processed data, we needed to determine the best features, and then the best algorithms that would enable us to perform forecasting on the data. After a brief overview of the literature, these major milestones outline the chapter's structure: understanding the data, selecting a storage platform, processing the data and performing analyses, and finally developing a simple visualisation for hoteliers.

5.1 Tourism Analytics

The emergence of smart tourism encompasses multiple data producers and consumers and is loosely defined as the growing trend for the reliance of tourism processes on ICT. Up to the millennium, this was focussed on Web 1.0 provision of end-user accessible material (typically brochures). The following decade was characterised by the growing influence of Web 2.0 in the context of interactivity and the emergence of consumer-driven ratings (e.g. [Xiang and Fesenmaier 2017; Gretzel et al. 2015a]). This technological step has produced growing data repositories that have great value in understanding the interplay between organisms in the tourism ecosystem [Gretzel et al. 2015b]. Analysis of these relationships supports the development of predictive informatics as a basis for influencing future activities.

Author's Draft

Initial work on predicting tourism behaviour was typically focused on time-series analysis within data that represents monthly arrivals. This produced a mean absolute percentage error (MAPE) around 4%, indicating the stability of the approach when applied to data aggregated on a monthly basis [Chu 2004]. The growth in Web usage has produced corpora such as Google Trends (GT) that could be expected to give insight into consumer behaviour in many contexts including tourism (Google Trends Labs). Indeed, aggregating Web search data has been shown to be a useful predictor of tourist arrivals in a number of localities with MAPE varying from less than 1% [Li et al. 2017] to 4.5% [Padhi and Pati 2017]. For example, a significant correlation was observed between Google search activity focussed on hotel accommodation and hotel bookings in Portugal [Dinis et al. 2017]. This work considered historical data related to the period from 2004 to 2012 which was aggregated on a monthly basis. There is a clear contribution of seasonal variation to the patterns observed. Combining Web search data for visitor bureau website traffic with time-series based prediction of hotel bookings provides a 15% improvement in the MAPE for weekly room booking predictions [Yang et al. 2014].

Predictive studies based on Web search data aggregated at weekly or monthly levels of granularity show distinct annual trends. However, the level of aggregation is not particularly helpful to day-to-day business decision making. In particular, these studies leave open the question of providing a viable prediction that could be used to guide hoteliers on a day-to-day basis concerning the likely demand for their products and thus staffing levels and best use of discounting to boost demand.

5.2 Making sense of the available Glasgow data

Data in smart cities is generated in two main ways: either automatically, through infrastructure (e.g. sensing equipment) in place around the city, or manually, i.e. by human operators, authorities or even citizens. This qualitative distinction typically differentiates the nature of the data in several ways. Infrastructure-generated data provides a steady (and typically high) frequency stream, allowing us to obtain fresh (up-to-date) information and to plan for storage and analysis since the volume of information is typically predictable. In contrast, human-generated data often suffers from fluctuations in availability, staleness and low volume but is often more aligned with business goals. Another important distinction is that infrastructure data is typically delivered in a machine-readable form, often with an associated Application Programming Interfaces (APIs) for direct querying of the data. Contrasting this, human-generated data is often provided in forms optimized for human reading, and in file formats that are more difficult to parse programmatically (e.g. XLSX, DOC, PDF etc). This is an important consideration, since machine-readable data often requires minimal transformation and pre-processing, and when it does, the code and

Author's Draft

algorithms required to do so are typically easy to produce. On the other hand, data that is not automatically generated often requires extensive manipulation to extract useful information elements and the algorithms required to do so can be complex and error prone, as there are no guarantees that all documents will adhere to similar formats or contain the same type of information at the same locations within the document (e.g. [Ruiz 2019]). Much web generated information can be considered semi-automatic: data about human activities such as web searching is often provided with automatic like APIs and data formats but is inherently human generated so open to noise introduced by human use of systems.

With this in mind, we examined a range of data sources both from our project partners and open data available on the web. The characteristics of the datasets are shown in Table 5.1.

Table 5.1 Project datasets and their main characteristics

Dataset	Format	Readability	Size	Description
Skyscanner flight enquiries	CSV	Very High	Very large	Queryable API for flight searches and redirects to airline websites for bookings. Supplied via GCMB.
LJ Forecaster Hotel Occupancy	XLS	Medium	Small	Monthly hotel occupancy reports. Supplied via GCMB.
GCMB Event listings	XLS	Low	Small	Monthly planned events (e.g. music, sports, conferences, arts) in Glasgow
Springboard Footfall	HTML	High	Small	Queryable API for pedestrian counts in Glasgow's main shopping streets. Supplied via GCMB.
Glasgow Life venue attendance	XLS	Good	Small	Monthly attendance data collected from all Glasgow Life facilities. Supplied via GCMB.
SQW Forecast dataset	XLS	Medium	Small	Data from a tourism forecasting report obtained

Author's Draft

				through consultancy firm. Supplied via GCMB.
IATA airport codes	CSV	Very High	Small	Open data from OKFN.
Cyclist/ Pedestrian counts in City Centre	CSV	Very High	Small	Obtained with a manual count in Sept. 2014. Glasgow city open data.
International Air Passenger Traffic Route Analysis	CSV	Very High	Small	Total number of passengers flying from Glasgow airport to other International terminals on a monthly basis. From Glasgow Airport - 2014
UK airport data	CSV	Very High	Small	Various reports relating to passenger volume (PAx) and aircraft movement at UK airports on a daily basis. UK CAA supplied.
Glasgow car park feed	JSON	Very High	Small but often	Real time feed of Glasgow car park occupancy. Glasgow open data.

From these datasets, we selected to proceed using the Footfall, Skyscanner and Hotel datasets only, all provided to us by GCMB. The rest were excluded for a variety of reasons: some sets were produced with considerable delay to be practically useful for predictions (e.g. two or three months after the reference timeframe), others were one-off data snapshots, and others were not time-series data or did not contain adequate temporal resolution to be useful. Still, we noted that the datasets we used are, by nature, incomplete. For example, we calculated that Skyscanner redirect data represents as little as approximately 15% of the total passenger volume at Glasgow Airport, by interpolating with the UK CAA official data, but it is far more detailed. Furthermore, as Skyscanner's market share varies globally it may give a biased sample of visitors - aspects that must be considered when reviewing data sources. Hotel data comes from 25 major hotels and does not include all Glasgow hotels and accommodation types (e.g. B&B). Footfall data does not cover all entries and exits into an area of interest and thus provides a limited view of pedestrians at the given locations, not trajectories. In the context of a smart city, it is almost impossible to

Author's Draft

capture all data pertinent to a problem. This data, however, can be thought to be representative of the tourism industry's "pulse" – measuring it should give us a reasonably good idea of the health of the tourism industry in the city. Next, we describe the three datasets in more detail, and provide some exploratory visualisations for these, which were helpful in assisting our initial understanding of the data we had available. As a software platform for our project, we used Microsoft Azure, since this was available through our GCMB partner. Azure provides a unified platform for structured and unstructured data storage and querying, which includes an environment on which to execute machine learning algorithms on the data. For the purposes of this quick exploratory process, we uploaded the available data into Azure storage, from which we could easily produce simple aggregations through SQL-type queries. The results of these queries were downloaded and imported into Microsoft Excel as .csv files, which we then used to produce quick visualisations.

5.2.1 Footfall data

Footfall data is obtained through automatic sensors installed in various locations around Glasgow using equipment from Springboard Ltd, based on real-time CCTV image analysis. There are 4 main locations of interests in the city centre pedestrian shopping streets (Argyle St., Buchanan St. (2 locations), and Sauchiehall St.) with two further locations (Sauchiehall St. and Bath St.) outwith the main pedestrian areas. Their approximate locations are depicted in Figure 5.1.

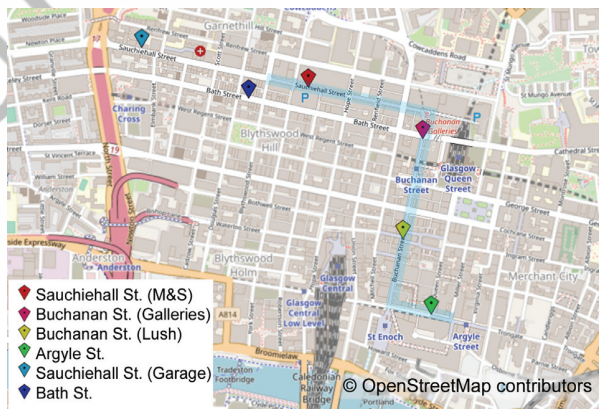


Figure 5.1 Location of the pedestrian footfall equipment. The main pedestrian shopping streets (the Style Mile) are marked in light blue.

The API provided allows querying using a start and end date parameter, and returns an HTML webpage with the sensor ID, date, hour and total pedestrian count observed in that hourly period. We downloaded the data for years 2014 and 2015, parsing the

Authors Draft

generated HTML with a simple script and inserted the data into a relational MySQL database table. This allowed us to perform basic querying to examine the data for seasonality and temporal patterns. In Figure 5.2, we can see some indications of a temporal pattern emerging across the two years (e.g. a repeating weekly spike pattern and a general seasonal trend). Such a detailed view of the data also gives insight to some special events: A marked increase of footfall at the end of July 2014 and the beginning of August 2014 (heavily influencing the 30-day moving average), and a very large spike on the 13th of September of 2014. These show respectively increases for two major events:

1. The Glasgow 2014 Commonwealth Games with approx. 690,000 visitors to the city[1].
2. The largest pro-independence rally[2] held in Glasgow five days before the 2014 independence referendum.

We also note the very distinctive drop in the Christmas and New Year week in both years. Some of these trends and events are likely to repeat and so should be modelled while other are one-offs - deciding how to handle these can be a challenge but is helped by working closely with tourism partners and adding complementary data of upcoming similar one-off events.

Author's Draft

[1] XX Commonwealth Games Visitor Study: Visitor Survey Results Report www.gov.scot/publications/xx-commonwealth-games-visitor-study-visitor-survey-results-report/

[2] www.bbc.com/news/uk-scotland-scotland-politics-29190306

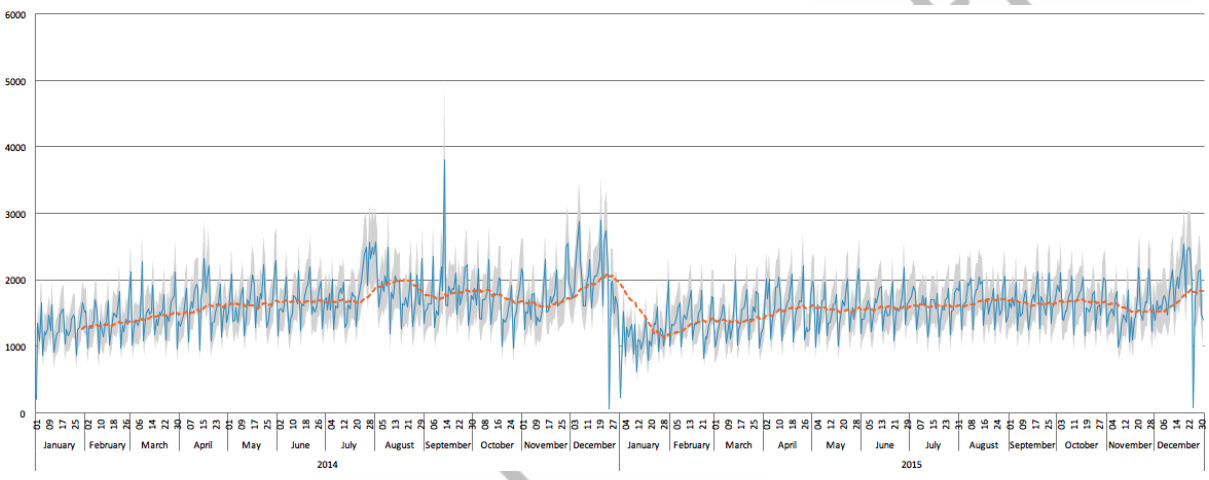


Figure 5.2 Daily footfall average by month and weekday across 2014-15 (error bands at 95%c.i.). Orange line presents a time-series moving average (30 days).

Better indication of a temporal pattern in the data comes when we aggregate the averages per weekday in each month for both the 2014 and 2015 data (Figure 5.3). We note here clearly that footfall in the city's shopping streets increases steadily Monday to Saturday, with a large drop on Sundays (as the shops are open for fewer hours and in Scotland that is traditionally a less shopping focussed day). The patterns are closely aligned across both years. This analysis gives us a fair indication that footfall in the city is likely predictable with reasonable accuracy.

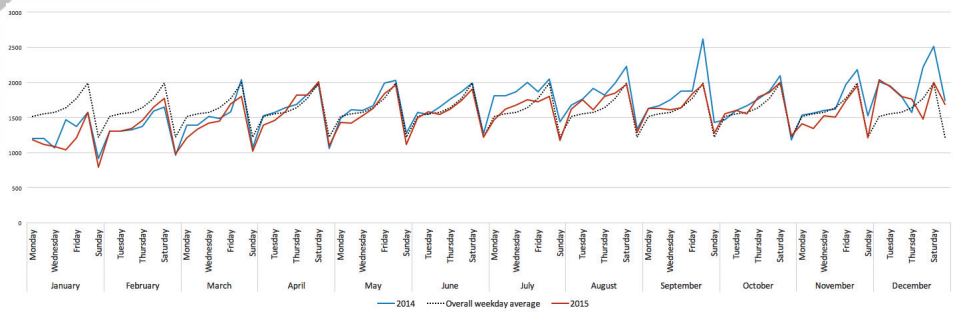


Figure 5.3 Footfall average by month and weekday in 2014 and 2015.

The analysis of footfall highlights the need to understand the nature of the underlying activity being monitored before blindly applying prediction: the special events and weekly cycle of a city are key to tuning models to accurately reflect city life.

Author

5.2.2 Hotel advanced reservation data

Hotel data was obtained from LJ Forecaster, a Scottish based tourism market research agency. The data contained room booking information on a daily basis for 25 hotels in the city centre and a further 8 hotels in the greater Glasgow area. The data is provided with a delay of approximately 1 month for the preceding monthly period, and includes the number of available rooms, the number of booked rooms, the number of "sleepers" (persons) staying in booked rooms and the generated revenue, for each date. The data is provided in Excel spreadsheets, which were parsed with a custom script and inserted in a relational MySQL database. Similar to footfall data, we plotted the booked rooms and sleepers both on a daily basis and aggregating by week (Figure 5.4, Figure 5.5). We also plotted the revenue figures (Figure 5.6).

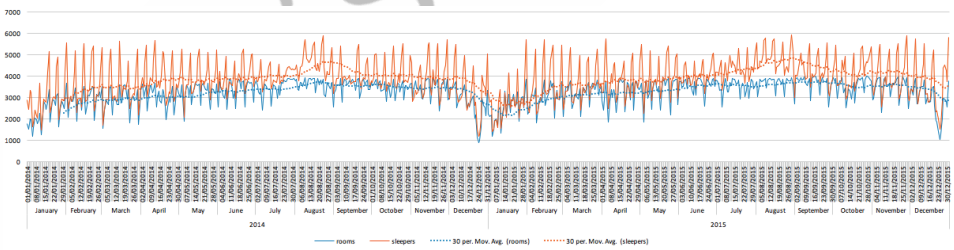


Figure 5.4 Daily hotel occupancy by week in 2014-15 (error bands at 95% c.i.)

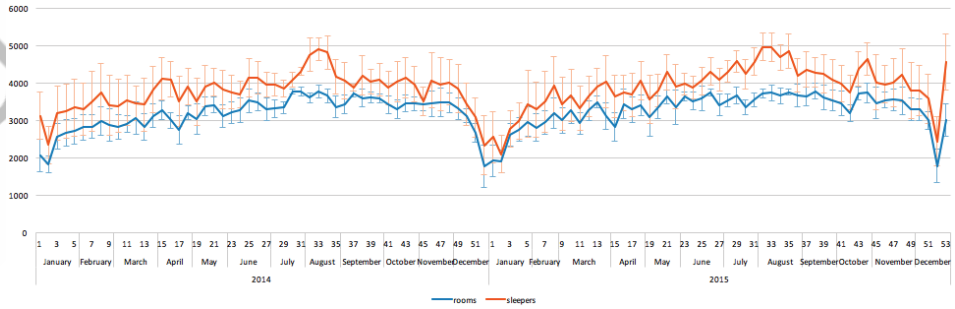


Figure 5.5 Hotel occupancy average by week in 2014-15 (error bands at 95% c.i.)

Author's Draft

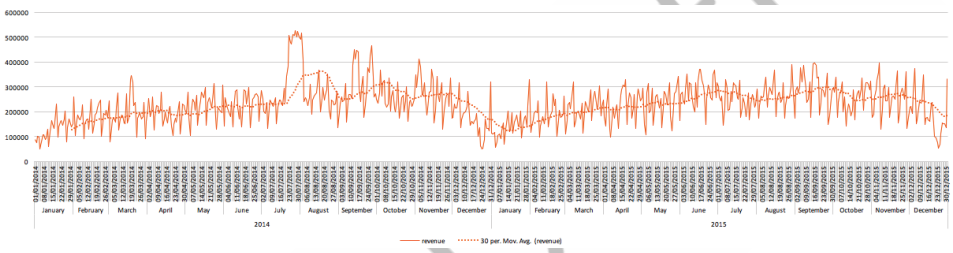


Figure 5.6 Daily hotel revenue by week in 2014-15

As before, we note that a regular temporal pattern emerges, with one notable event, i.e. a large spike in revenue coinciding with the Commonwealth games held in the city in July 2014. Surprisingly this peak is not accompanied by a similarly high peak in room bookings or total sleepers, showing how the market can work in such exceptional circumstances.

5.2.3 Skyscanner Flight Search Data

Data in the Skyscanner set is separated in two distinct subsets: searches and redirects. Searches is a log of all searches for flights to Glasgow International (GLA) on the website and mobile app, Redirects is a smaller dataset containing details of searches that were redirected to airline or travel agent websites so that the customer could book a selected flight. At the time Skyscanner did not offer integrated flight bookings, but only information so that the user could be redirected to the relevant website to complete a booking. Therefore while "redirect" should be viewed as a strong indicator of a flight booking, it is not a guarantee that the booking was actually completed. The logs offer a wealth of data fields including intended travelling dates (inbound and outbound flights), the number of seats (adults + children), the starting airport for the search and the user's local airport. Due to the massive file size, the data (provided as CSVs through a queryable API) was held in Microsoft Azure DataLake storage, which allows querying through T-SQL.

We first attempted to estimate the visitors present in Glasgow on a given date based on the search and redirect data. We excluded records for day visits (outward = return date), trips with a duration was more than 10 days (based on advice from GCMB as these are often students or long-term visitors who want to tour Scotland outside Glasgow), and one-way trip records (no return requested). Following this filtering, we estimated the number of visitors bound to be in the city for each given day in 2014 and 2015, by treating each search as being made by a single and distinct individual and tallying up the number of individuals (seats) who have declared that they would be in Glasgow (by stating an inbound and outbound date). As an example, Table 5.2 shows two searches with associated occupancy estimates.

Author's C

Table 5.2 Sample flight search data

				Occupancy			
Search ID	Seats	Inbound	Outbound	1 Jan	2 Jan	3 Jan	4 Jan
10302	1	01/01/2015	03/01/2015	1	1	0	0
10303	2	02/01/2015	04/01/2015	0	2	2	0
				1	3	2	0

By applying these calculations to the search dataset and the redirect dataset, we produce two metrics of busyness:

- Likely Visitors: number of visitors present in the city calculated from the redirect dataset
- Interested Visitors: number of visitors present in the city calculated from the search dataset

The supplied data did not explicitly link searches and redirects nor do they identify individuals or machines making the searches, so we have no way of identifying whether two or more search / redirect records have been made by the same individual. Hence, these two metrics are a rough approximation and are a combination of strength of interest (a person might be repeatedly searching for the same dates) and actual intended visits. Therefore we have a total of four busyness indicators:

- Likely Arrivals: Number of people flying into Glasgow on a given date (redirects)
- Likely Visitors: Number of visitors from any trip in Glasgow on a given date (redirects)
- Interested Arrivals: Number of people flying into Glasgow on a given date (searches)
- Interested Visitors: Number of visitors from any trip in Glasgow on a given date (searches)

The calculated figures for each date were stored in a relational MySQL database for further processing. A timeline of these metrics is shown in Figure 5.7.

Author's Draft

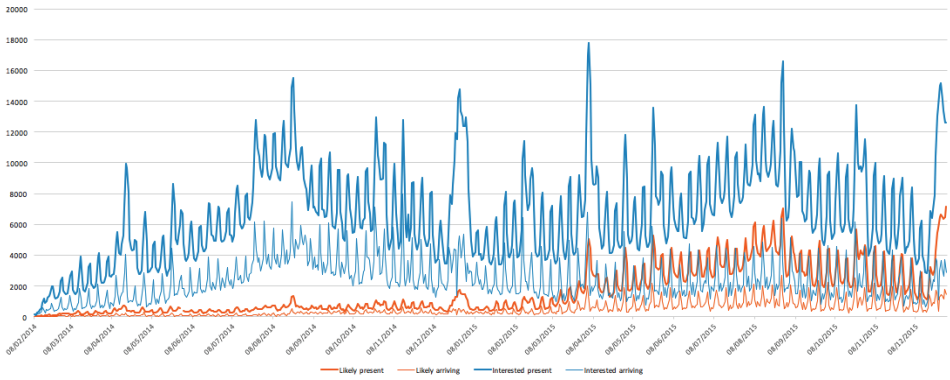


Figure 5.7 Daily visitors presence and arrivals in 2014-15

From this data, we note a regular (weekly) pattern showing increased interest in weekend arrivals (Friday & Saturday) as well as a seasonal trend, with increased traffic in the summer months (July and August). We note here that there doesn't seem to be much correlation between searches and redirects prior to April 2015. This appears to have been a result of Skyscanner changing how redirects were promoted from search results. It is also worth noting that interest for Glasgow in the Skyscanner platform is quite low in early 2014. This is indicative of their early growing market share combined with searches for future travel. Therefore, in contrast with other datasets (hotels & footfall), it is important to note here that the observed figures are as reflective of the Skyscanner platform's market success and interface design changes, as they are of interest in the city of Glasgow.

A user interface related issue also became clear on analysing the data: the first of the month appeared to get a boost, as did travelling on the day of the search. Same day bookings are very rare and GCMB did not believe the first to be particularly significant, however in the interface if users do not set a date it defaults to travel today and we hypothesize that users doing speculative searches may pick, say 1st June to get an idea for summer trip prices to Glasgow with no plan to book immediately.

Author Draft

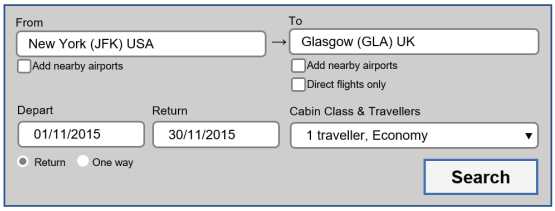


Figure 5.8 Sample flight search interface c. 2015

5.3 Predicting business indicators

Although our initial data exploration used unsophisticated tools, for the purposes of producing the desired predictions, we needed to employ more powerful tools. For this purpose, we selected, amongst a range of options, to explore the data using the R language. R is a widely popular platform for statistical computing with many packages for statistical and machine learning approaches, and is also supported by Azure. As we describe our findings in the next sections, we will also mention the R packages and tools used throughout the analysis.

Since, as can be seen, our data consists of different timelines with varying granularity, there are two approaches that could be used to predict the key business indicators from this data. The first approach is to use each data series individually and to try to predict future values based on the history of the time series (univariate approach). This approach may yield good results, particularly when the data exhibits a clear temporal pattern (i.e., periodicity). On the other hand, this approach misses the interplay between different data sources, and therefore combining data sources into a single predictive model, may yield far more accurate results.

5.3.1 Univariate models

We performed univariate timeseries predictions on our datasets: can we can predict the value of a given metric, based on this metric's history alone. For this type of analysis, as well as for the multivariate analyses that follow, we used the libraries offered by the R *'forecast'* package[3]. For univariate analysis, we applied four models, which are explained in great detail (both theory and use in R) in [Hyndman and Athanasopoulos 2018]:

- **STL-naive**: Seasonal Trend decomposition using Loess) and a naive predictor [Hyndman & Athanasopoulos 2018, ch. 6]
- **HWA**: Holt-Winters seasonal method (additive model) [Hyndman & Athanasopoulos 2018, ch. 7]

[3] R Forecast package https://github.com/robjhyndman/forecast

Author's Draft

- **HWM:** Holt-Winters seasonal method (multiplicative model) [Hyndman & Athanasopoulos 2018, ch. 7]
- **NNAR:** Neural Network Autoregression [Hyndman & Athanasopoulos 2018, ch. 11]

The time series in all cases are considered to have a seasonal parameter of 7 (i.e. one week). We consider all data starting from Week 0 (beginning of our datasets) to Week 95 as inputs to the model's training. From this data, we predict the values for the 21 days (3 weeks) following the end of Week 95. While these models are tunable through a range of parameters, in this initial investigation we assessed their predictive performance without investigating the optimal tuning of these parameters and instead use a set of parameter values that are reasonable estimates based on the nature of the data in the time series. The parameters are shown in Table 5.3.

Table 5.3 Parameters for univariate modelling

Model	Parameter	Value
STL	Trend window	30
	Seasonal window	7
	Robust	TRUE
	Data points to predict	20
HWA	Seasonal model	"additive"
	Data points to predict	20
HWM	Seasonal model	"multiplicative"
	Data points to predict	20
NNAR	Seasonal lags	4
	Hidden nodes	5
	Number of NNs to train	20
	Data points to predict	20

Since in this project we were mostly interested in examining the predictive power of the combined datasets, we won't report on all the outcomes of the univariate analyses here, but since this technique can be useful in providing at least a form of baseline on

Authors Draft

the predictive power of each individual time series, we will work through one example of our key prediction targets: hotel room and sleeper bookings.

5.3.2 Univariate predictions on hotel data

We start off with STL, as this technique attempts to break down a time series into two components that can provide valuable insights about a time series. These are the seasonal and trend components. As can be seen in Figure 5.9, the hotel room bookings demonstrate a steady seasonal, in our case weekly, variation while the trend pattern here shows how hotel business varies throughout the years. The "remainders" at the bottom of the figure are useful in identifying departures from "normal" values, i.e. extreme cases (such as the large dip just after the New Year period). With such clear seasonal and trend components, we can have at least some confidence that this time series has reasonable predictive power.

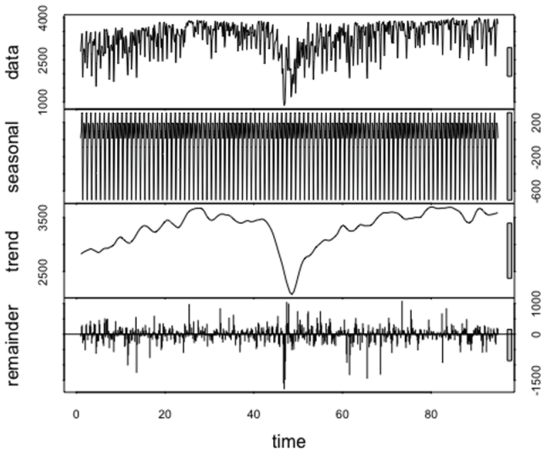


Figure 5.9 STL decomposition results on hotel room booking data (time is in weeks)

Running the forecasting models on the hotel booking data time series, gives mixed results. We can see that all models perform well in predicting the highs and lows of the dataset, but not that well in the "in-between" values. A comparison of the average prediction error shows that overall, for the 3 weeks, the models performed with an average error rate between 10-15% of the actual level of room bookings for all the dates in the prediction period. This is a fairly good start and sets the baseline for any further improvements through combining data. In the next section, we will discuss how combining multiple time series affected the prediction targets.

Author's Draft

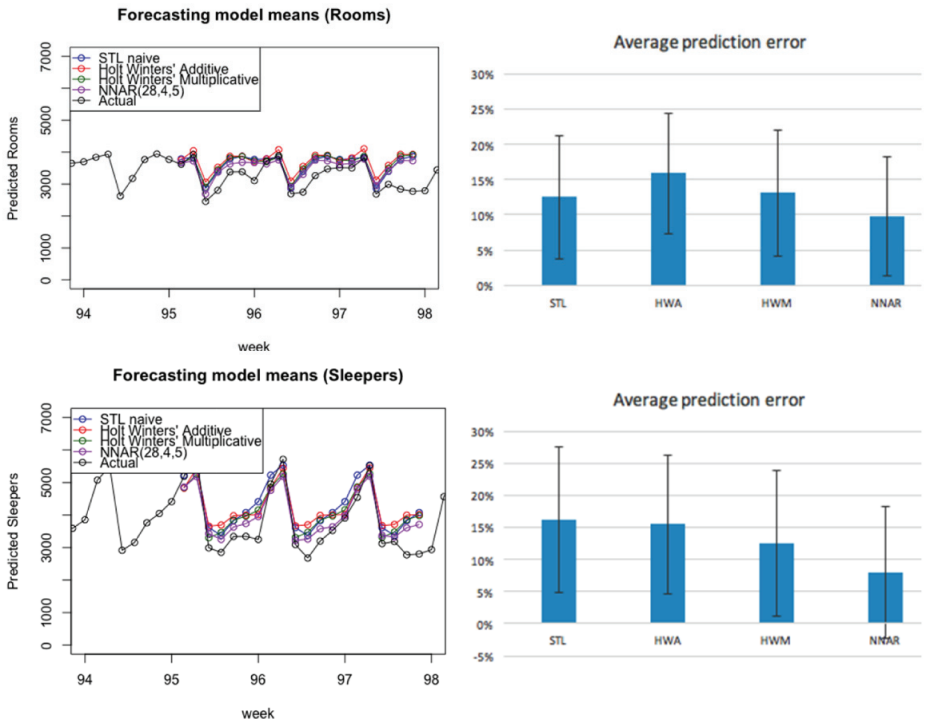


Figure 5.10 Univariate predictions for hotel room (above) and sleepers (below)

Overall, using a 90 day training window and running the analysis for the entire July'15 - May'16 period (predicting 4 weeks ahead of the first of each month), we see in one case we are able to obtain a reasonably good accuracy (STL, sleepers), but overall the error margins are quite high (Figure 5.11).

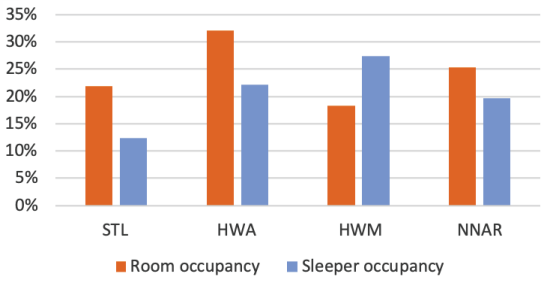


Figure 5.11 NRMSE for univariate predictions

5.4 Multivariate predictions

Multivariate analyses can yield potentially more accurate results by capturing the relationships between a high number of features, but may result in an overly complex and computationally intensive model. Sometimes, adding more features in a dataset doesn't necessarily mean that we will improve the prediction performance, and a reason for this is often that data features may exhibit strong correlations. If two features are strongly correlated, we gain very little new information by including both features in a predictive model. Therefore, we should attempt at the start to examine our data features for correlations and to try to reduce the dimensionality of our dataset as much as possible.

5.4.1 Multivariate features

In our case, our dataset contained 14 features (Table 5.4), some from raw data and some derived from the raw datasets. All values are calculated on a daily basis.

Table 5.4 Features for multivariate prediction

Feature	Description
Date	The date for which the feature values are calculated
Likely Visitors	Number of visitors present in the city calculated from the redirect dataset
Rooms	Sum of booked rooms in all hotels
Sleepers	Sum of booked sleepers in all hotels
Revenue	Revenue generated from all hotels
Interested visitors	Number of visitors present in the city calculated from the searches dataset
Footfall	Average number of pedestrians detected per location
Average rooms	Average number of room occupancy per hotel
Average sleepers	Average number of sleeper occupancy per hotel
Likely Arriving trips	Number of trips arriving into Glasgow, calculated from the redirect dataset
Likely Arriving passengers	Number of passengers arriving into Glasgow (each "trip" can have multiple passengers), calculated from the redirect dataset

Author's Draft

Interested trips	Number of trips arriving into Glasgow, calculated from the searches dataset
Interested arriving passengers	Number of passengers arriving into Glasgow (each "trip" can have multiple passengers), calculated from the searches dataset
Total trips	Sum of interested and likely trips
Total arriving passengers	Sum of interested and likely passengers

As noted earlier from inspection of the data we saw that there are two periods of stabilisation of the data: From July 2014 onwards, the number of total arriving trips seems to stabilise. However, the total number of passengers increases heavily from April 2015 and then stabilises in a regular pattern of volume. This means that prior to that date, users were probably not declaring an accurate number of passengers (which makes sense, because we see that prior to that date, the website was used mostly for searches). Once redirects begin to overtake the volume of searches, the volume of declared passenger increases. This means that we should, for correlation purposes, only consider Skyscanner data after 01/04/2015. The other data series (hotels, footfall) are regular and do not need the application of exclusion criteria.

5.4.2 Feature correlation analysis

Based on this exclusion, we can investigate possible correlations between pairs of features. One way to achieve this visually, is by plot the correlations and scatterplot matrices (Figure 5.12), showing combinatorial scatterplots of all variable pairs. These are easy to produce, using the *scatterplotMatrix()* function from the *'car'* R package, and the *corrgram()* function from the *'corrgram'* package, and providing in both cases only the dataframe as the sole parameter..

Authors Draft

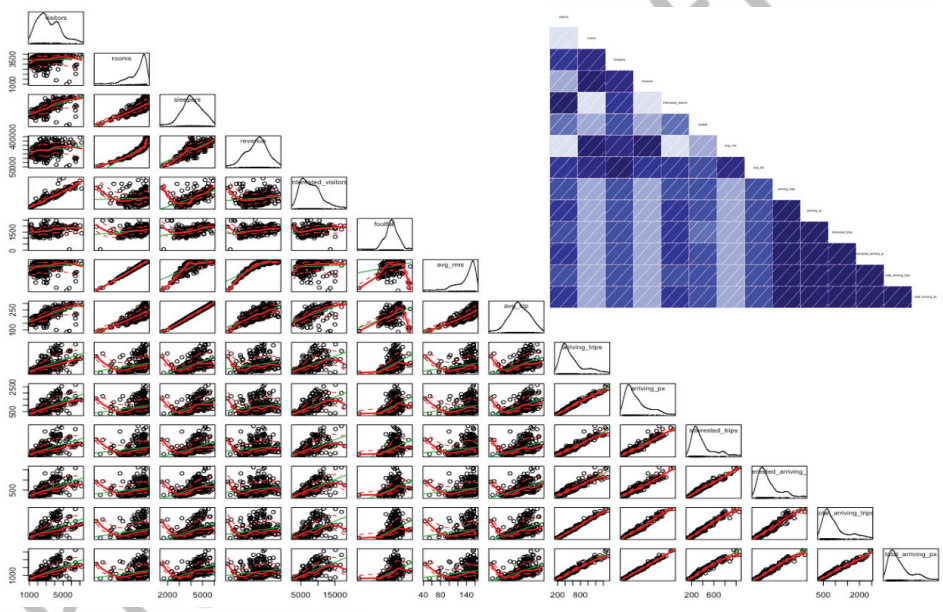
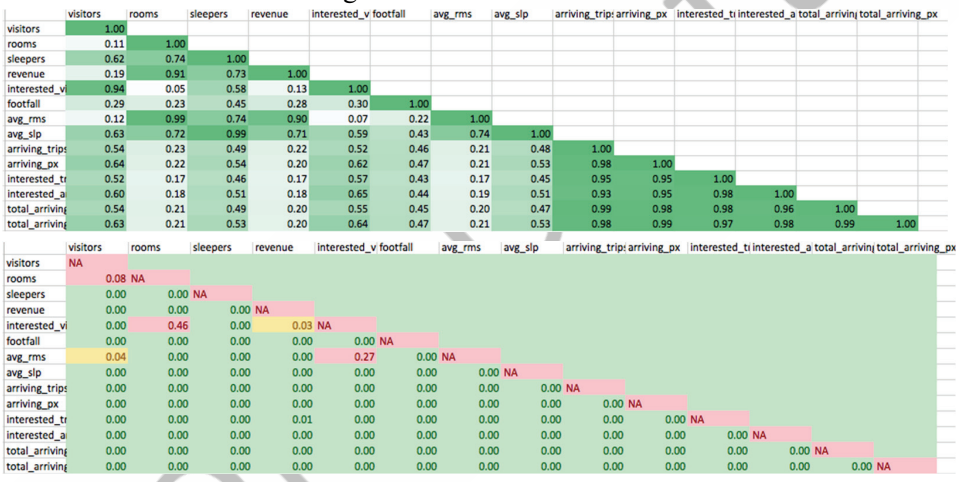


Figure 5.12 Scatterplot matrix with correlations matrix insert

In addition to visually inspecting the correlations matrices it is essential to calculate the level of correlation between each pair and to test for statistical significance. We calculated correlation using Spearman's Rho as shown in table 5.4.2. For this, we used the *rcorr()* command from the *'Hmisc'* package, which calculates all pairwise correlations and their statistical significance, and provides them in a handy table format.

Table 5.4.2 Multivariate correlation and significance tables



	visitors	rooms	sleepers	revenue	interested_v	footfall	avg_rms	avg_slp	arriving_trips	arriving_px	interested_t	interested_a	total_arriving	total_arriving_px
visitors	1.00													
rooms	0.11	1.00												
sleepers	0.62	0.74	1.00											
revenue	0.19	0.91	0.73	1.00										
interested_v	0.94	0.05	0.58	0.13	1.00									
footfall	0.29	0.23	0.45	0.28	0.30	1.00								
avg_rms	0.12	0.99	0.74	0.90	0.07	0.22	1.00							
avg_slp	0.63	0.72	0.99	0.71	0.59	0.43	0.74	1.00						
arriving_trips	0.54	0.23	0.49	0.22	0.52	0.46	0.21	0.48	1.00					
arriving_px	0.64	0.22	0.54	0.20	0.62	0.47	0.21	0.53	0.98	1.00				
interested_t	0.52	0.17	0.46	0.17	0.57	0.43	0.17	0.45	0.95	0.95	1.00			
interested_a	0.60	0.18	0.51	0.18	0.65	0.44	0.19	0.51	0.93	0.95	0.98	1.00		
total_arriving	0.54	0.21	0.49	0.20	0.55	0.45	0.20	0.47	0.99	0.98	0.98	0.96	1.00	
total_arriving	0.63	0.21	0.53	0.20	0.64	0.47	0.21	0.53	0.98	0.99	0.97	0.98	0.99	1.00

	visitors	rooms	sleepers	revenue	interested_v	footfall	avg_rms	avg_slp	arriving_trips	arriving_px	interested_t	interested_a	total_arriving	total_arriving_px
visitors	NA													
rooms	0.08	NA												
sleepers	0.00	0.00	NA											
revenue	0.00	0.00	0.00	NA										
interested_v	0.00	0.46	0.00	0.03	NA									
footfall	0.00	0.00	0.00	0.00	0.00	NA								
avg_rms	0.04	0.00	0.00	0.00	0.27	0.00	NA							
avg_slp	0.00	0.00	0.00	0.00	0.00	0.00	0.00	NA						
arriving_trips	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	NA					
arriving_px	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	NA				
interested_t	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	NA			
interested_a	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	NA		
total_arriving	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	NA	
total_arriving	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	NA

The correlation analysis shows that we can derive significant relationships between almost all the features in the dataset. An interesting observation is that while the calculated number of visitors in the city on any given date (as derived from searches and from redirects) does not correlate with rooms booked on that date, a strongly significant correlation exists with calculated trip arrivals and passenger arrivals on the date. The underlying assumption for this derived metric is that people flying into Glasgow will be staying in the city, and therefore will be booking hotel rooms for their stay. However, the lack of correlation shows that this assumption might be wrong. Many tourists use Glasgow airport as a gateway for further travel into the country (e.g. to the Highlands or to Edinburgh). Furthermore in discussing this data with the tourism office, they highlighted that around 30% of arrivals stay with friends or family. An additional concern could be that our hypothesis is not incorrect, but the way we are estimating present visitors might be wrong (e.g. the 10 days cut-off window is too large and the average stay is much shorter). However, we cannot verify these explanations with our given data.

5.4.3 Reducing the complexity of our dataset: dimensionality reduction

Because of the many correlations, it makes sense in our case to implement some method of dimensionality reduction, in order to avoid “feeding” data into our predictive model which doesn’t contribute significant new information. Applying dimensionality reduction allows us to create simpler and faster models (both for training and execution purposes), without losing (and indeed, sometimes improving) the quality of predictions. It also helps with storage requirements, and to understand and visualise the data (especially if they can be reduced to two or three dimensions).

Author Draft

There are several dimensionality techniques that can be applied to reduce the complexity of a dataset, and here we chose the Principal Component Analysis (PCA) technique. In our case, because of the many correlations in the data features, we expect that PCA will be able to significantly reduce the dimensionality of our dataset, resulting in far fewer features to be used for modelling and prediction. In PCA, a principal component is a linear representation of the combined data values, multiplied by a weight w (loading factor), in general: w_a*a + w_b*b +..+ w_n*n . The loading factors represent the correlation between the principal components and the variables. The process relies on the assumption that multiple observed values have a similar pattern, because they are all associated with some latent concept, which we can't directly measure (or perhaps didn't think to measure!). Therefore each principal component represents this underlying latent concept.

The first principal component derived by PCA is the linear combination of variables that has maximum variance (among all linear combinations), so it accounts for as much variation in the data as possible. The second principal component is the linear combination of variables that accounts for as much of the remaining variation as possible, with the constraint that the correlation between the first and second component is 0 (and so forth, until the i^{th} PC is calculated and all variance has been accounted for). The loading factors are computed from the eigenvalues & eigenvectors of the Variance-Covariance matrix of our variables (data).

Starting off with PCA, it is crucial to remember that this technique is a scale-dependent process, hence, we should first normalise the data, which can be easily done using the *scale()* function, part R's core packages. Once this is done, we can derive a "scree plot" to determine how many factors to keep (Figure 5.13). To do this, we use the *prcomp()* function, part the R core packages. The results of this function can be used as a parameter for the *screeplot()* function, also part of the R core packages.

Author's Draft

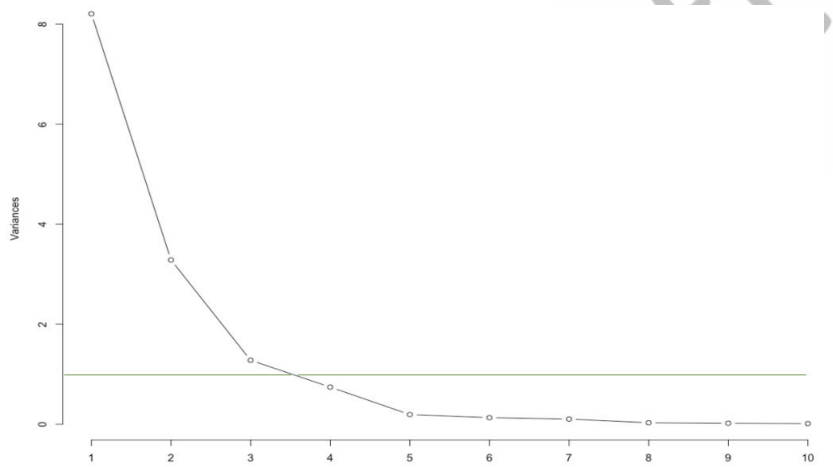


Figure 5.13 Scree plot: Principle Components and their component wise variance

In a scree plot, we're looking for the "elbow" in the plot, which happens in principal component 5. This elbow shows the point after which the gain in being able to explain additional variance begins to become insignificant. While there are other approaches (e.g. [Rea and Rea 2016]), Kaiser's rule dictates that we should keep only components with a variance >1, hence we should retain just the first 3 components. In our case, we can consider the application of the Kaiser criterion to be sufficient, as the first 3 components also explain 91.2% of all variance, as we can see from Table 5.4.2. The values are obtained from the result of the *prcomp()* function.

Table 5.4.2 Contribution of each principal component

Principal Component	Component-wise variance	Cumulative Variance
PC1	8.18	0.59
PC2	3.28	0.82
PC3	1.28	0.91
PC4	0.74	0.97
PC5	0.19	0.98
PC6	0.13	0.99

PC7	0.10	1.00
PC8	0.03	1.00
PC9	0.02	1.00
PC10	0.01	1.00
PC11	0.00	1.00
PC12	0.00	1.00

Next, we should examine the loadings of the variables in each of the principal components (Table 5.4.3), again given by the results of the *prcomp()* function. This can give an interpretation of how the principal components are related to our original values. Explaining the principal components in terms of their values is a subjective process in which we look at those variables which have the highest loading factors (absolute values). However, the groupings of the variables must make theoretical sense to the researcher, i.e. that the highly loaded variables group logically together in the context of the dataset. Here for the first 3 PCs we can say that the first PC is a measure of the Skyscanner data and to a lesser extent hotel sleepers. Remember our previous explanation that not all people who fly into Glasgow actually stay in hotels? PC2 is a measure of our hotel data (revenue and rooms) and to a lesser extent sleepers. This also makes logical sense, since the revenue is mostly dependent on the number of rooms sold, and not the number of people staying in each room (in other words, often the price difference between a single and double occupancy room is quite small). Finally, PC3 is a measure of the interested visitors and likely visitors from our processed skyscanner data. As we can see here, footfall doesn't seem to play a role in accounting for variance in any of the factors.

Authors Draft

Table 5.4.3 Breakdown analysis of first three components.

PC1		PC2		PC3	
Total arriving passengers	**0.326**	Interested arriving trips	0.195	Arriving trips	0.227
Arriving passengers	**0.325**	Interested arriving passengers	0.183	Total arriving trips	0.220
Interested arriving passengers	**0.318**	Total arriving trips	0.179	Interested arriving trips	0.202
Total arriving trips	**0.317**	Total arriving passengers	0.176	Booked rooms	0.185
Arriving trips	**0.315**	Arriving passengers	0.168	Average room occupancy	0.158
Interested arriving trips	**0.310**	Arriving trips	0.167	Arriving passengers	0.122
Booked sleepers	0.274	Interested present visitors	0.110	Total arriving passengers	0.119
Average sleeper occupancy	0.272	Present visitors	0.074	Interested arriving passengers	0.109
Present visitors	0.251	Total footfall	-0.021	Average revenue	0.089
Interested present visitors	0.248	Average sleeper occupancy	-0.285	Total footfall	0.086
Total footfall	0.190	Booked sleepers	-0.287	Booked sleepers	-0.212
Average revenue	0.165	**Average revenue**	**-0.446**	Average sleeper occupancy	-0.232
Average room occupancy	0.164	**Average room occupancy**	**-0.463**	**Interested present visitors**	**-0.567**
Booked rooms	0.164	**Booked rooms**	**-0.463**	**Present visitors**	**-0.568**

5.4.3 Predicting key business indicators using statistical modelling

Before going into more complex modelling techniques, it's important to know that we can directly use the results of the preceding PCA process to perform a regression on our data (PCAR). Using the first 3 PCs as identified above, we can try to predict for example, the number of booked rooms. This is done by a simple linear combination of the identified weights as follows:

rooms = 0.326 total_arriving_px + 0.324 arriving_px … + 0.164 rooms + 0.195 interested_trips + 0.183 total_arriving_px … - 0.463 rooms + 0.227 arriving_trips + 0.220 total_arriving_trips … - 0.568 visitors

To perform PCA-based regression in R, we can use the *pcr()* function of the *'pls'* package. In Figure 5.14 we use the first 120 rows (starting 01 April 2015) as a training sample on which to perform the PCA and then perform predictions up to 31/12/2015 using 3 principal components, for booked rooms. The orange line shows predictions, while the blue line shows actual values. Figure 5.15 gives another example for arriving visitors, and Figure 5.16 shows the same example (arriving visitors) with a larger training set (predicting only the last 35 days). As we can see, the results are quite promising.

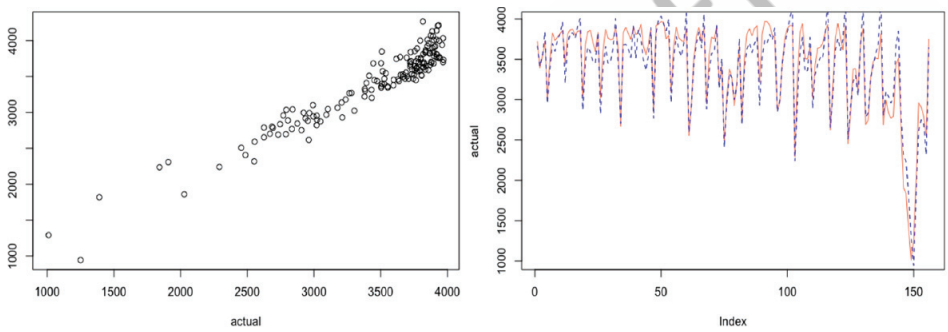


Figure 5.14 Prediction of booked rooms

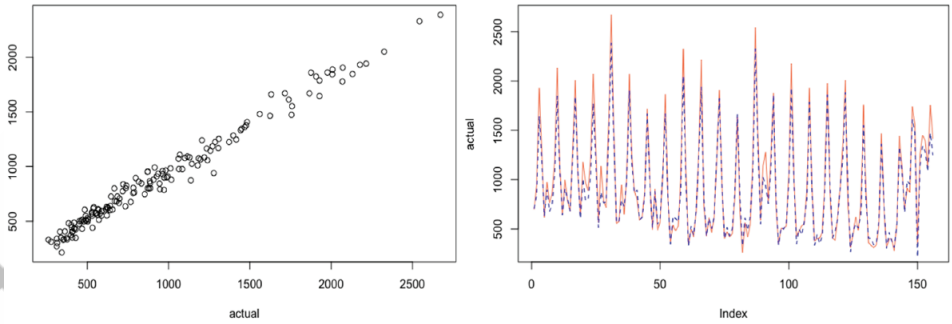


Figure 5.15 Predicting arriving visitors

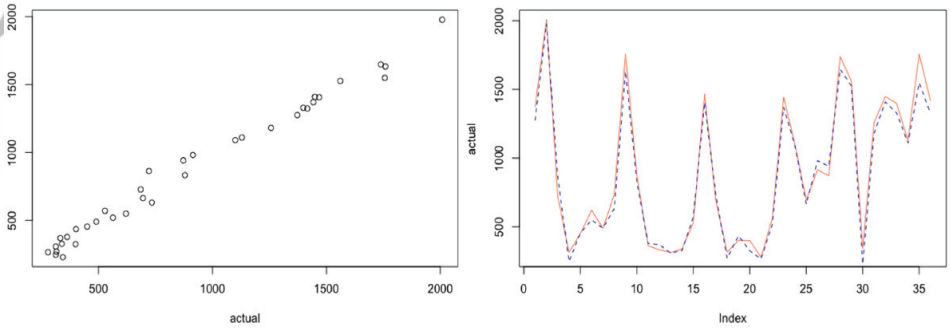


Figure 5.16 Arriving visitors ("searches" data set)

To investigate further, we implemented a "rolling" PCAR technique, in which we try to predict an entire month's values for each dataset feature, based on a training window of that includes the previous *n* months of data. For example, if the training window is 2 months, we can predict the values for March based on the preceding

January and February data, then we move on to predict April based on February and March data, and so forth. For this process, we examined different sizes for the training window. The results can be seen in Figure 5.17 below, where the normalized root mean square error is calculated for each feature (we implemented the formula for NRMSE in R ourselves, using the results of the *pcr()* function).

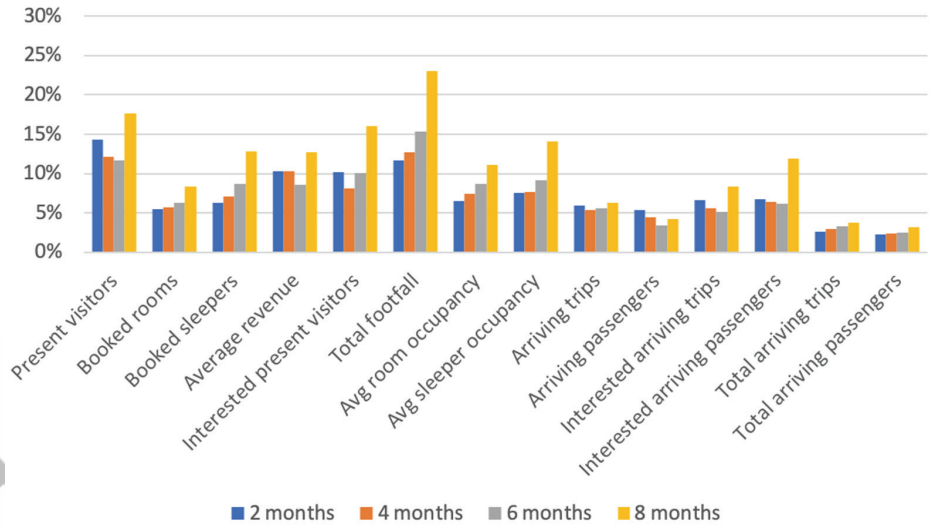


Figure 5.17 Effect of training window in PCA-R results on individual features

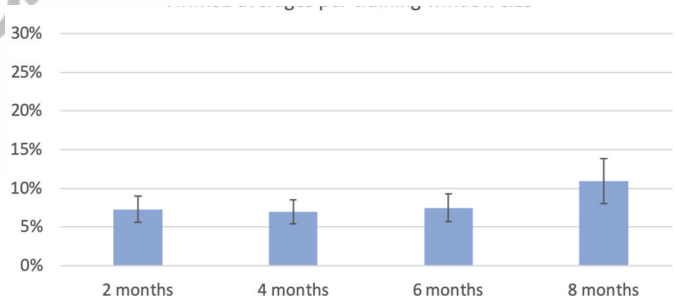


Figure 5.18 NRMSE average across all features per training window size (error bars at 95% c.i.)

As can be seen, although prediction performance is not the same for all features (i.e. some are predicticted more accurately than others), more training data does not mean better prediction performance. Nominally, the best performance across all features is attained with a training period of 4 months, but there is no statistically significant difference in the overall average NRMSE between the 2, 4 and 6 month training

windows (paired sample t-tests, with post-hoc Bonferroni correction setting the p-value threshold at $p<0.0083$). However, we find a statistically significant difference in the NRMSE between the 8-month training window length and all others (2m – 8m $t(13)=-4.236$, $p=0.01$, 4m – 8m $t(13)=-4.814$, $p<0.01$, 6m – 8m $t(13)=-5.535$, $p<0.01$). As a result, we notice that the rather good performance of ~7% NRMSE is obtainable with just 2 months of training data. More specifically, since the most crucial business metrics are the number of booked rooms, sleepers and revenue, the average NRMSE for these three using 2 month training windows over the entire year is 5.47% (booked rooms), 6.30% (sleepers) and 10.27% (average revenue).

5.4.4 Predicting key business indicators using machine learning

Neural networks have recently had explosive growth in use in machine learning and data analytics. In particular the neural network autoregression (NNAR) technique is a promising candidate for solving forecasting problems where the patterns and relationships in the data are not immediately obvious. To examine the predictive performance of NNAR, we used with the same variables that were previously used in the principal component analysis in NNAR modelling: Parameter, Explanation, Interested visitors, Visitors, Arriving trips, Arriving visitors, Interested arriving trips, Interested arriving visitors, Room occupancy, Sleeper occupancy, and Average revenue. Again we use the *nnetar()* function of the *'forecast'* package, only this time, we provide more than one feature as input.

At first, we attempted to predict each of the variables in turn from the remaining variables. Data was used in the range 01/04/2015 - 24/06/2016, since we had more data available to our project at that time. For each month (starting at 07/2015), the previous three months was used as a training set and predictions were performed on a day-by-day basis. We used a neural net structure with two hidden layers (of five and three nodes), the model was converged 30 times and the NRMSE error rates, averaged across all 30 runs, were calculated. An example of one generated model is shown in Figure 5.19 with output of the predictive performance of the model in figure 5.20 for both cities.

Authors Draft

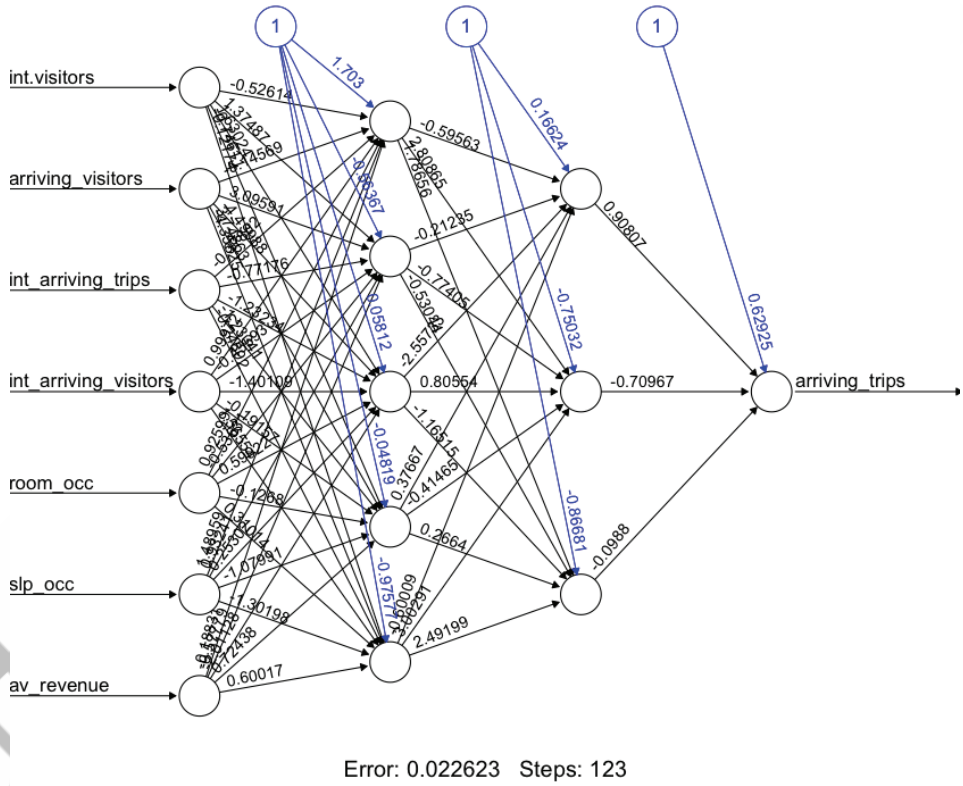


Figure 5.19 Example NNAR model

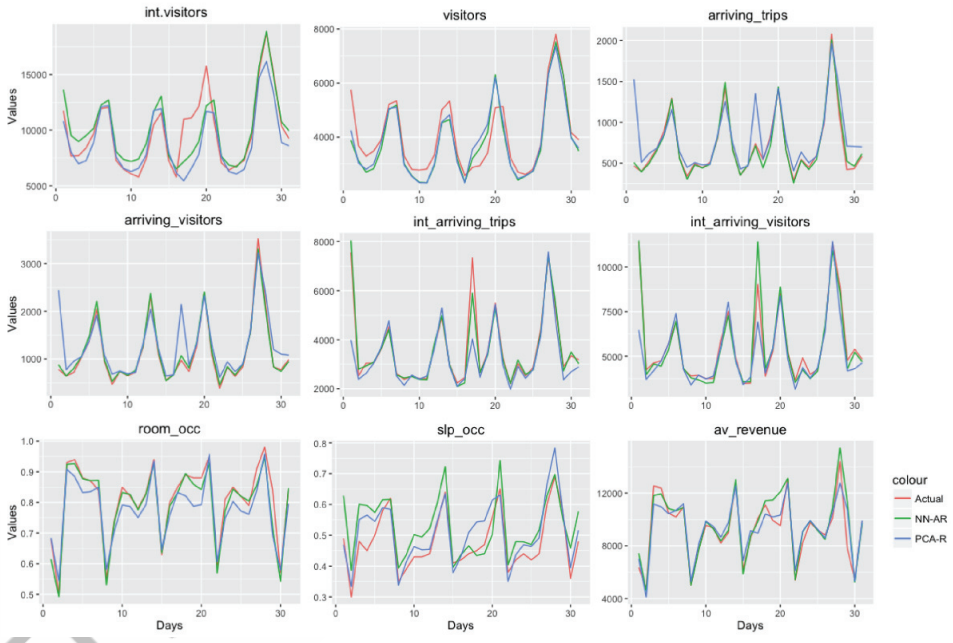


Figure 5.20 NNAR Modelling for Glasgow May 2016 using 3 months training data

Overall, we note that for the dataset in the time period examined, the NNAR method distinctly outperforms PCAR in seven of the nine features examined, while for the other two, the performance is comparable (Figure 5.21).

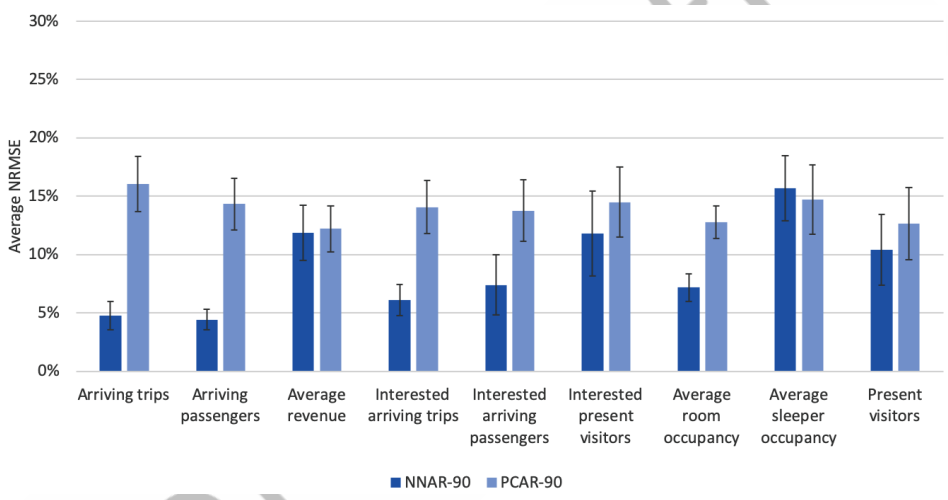


Figure 5.21 NNAR and PCAR prediction NRMSE averages (error bars at 95% c.i.)

We also compared the performance of the multivariate NNAR method on our two key performance indicators (room occupancy and sleeper occupancy) with the univariate NNAR (i.e. performance based on the historical values of each feature). From this analysis (Figure 5.22), we note that the multivariate NNAR offers consistently better performance compared to univariate (self) NNAR, considering the NRMSE, RMSE and absolute MSE (%).

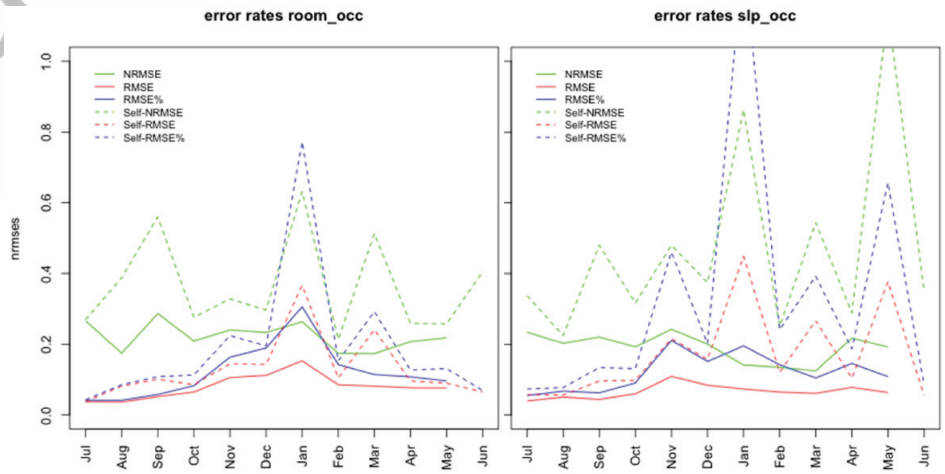


Figure 5.22 Multivariate vs. univariate NNAR performance for Glasgow

In the preceding analyses, there is an underlying assumption that all of the required training data is available at the time of training and running the models. However, in the real world, this is not always feasible. For example, the data offered by Skyscanner are real-time datasets and are available on-demand, since they are automatically generated through the visitor interactions with the website. On the other hand, hotel data is contributed manually by the hotel owners, collected, cleaned and aggregated off-line at the company that provides them, and therefore is often available with a considerable lag (at least one month, and in cases up to two months). Therefore, working with our dataset effectively only provides what can be considered a theoretical "maximum" forecasting ability, which is achievable only when all the required data is available. A far more interesting comparison thus, would be to use the real-time (Skyscanner) dataset as the sole input into a predictive model, with the aim of forecasting the hotel data, or vice-versa, assuming the hotel data was available in real-time and flight data was not. For this, we carried out a further experiment using data for the city of Glasgow, and attempted to predict using 3 months' worth of training data. In Figure 5.23, we show the comparative performance of this prediction for the month of May, across all variables. As we see, the predictive performance of the model suffers significantly, but it appears to be far easier to predict hotel data from flight data, than the other way round.

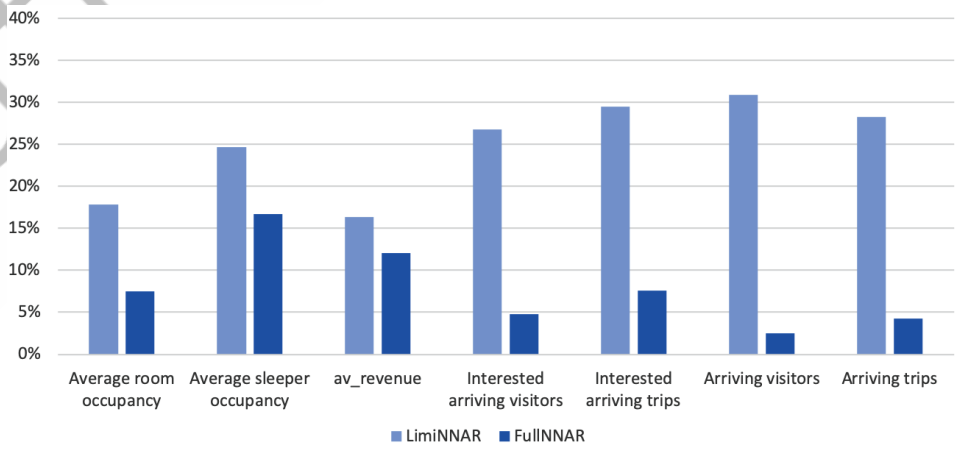


Figure 5.23 NRMSE per variable per input set (NNAR-90, May 2016) using Skyscanner only

Since our main prediction target is hotel occupancy figures, we looked at how the NRMSE varies across the whole July'15-May'16 dataset for these two features. As we can see, while performance with features from both datasets remains more or less constant, it suffers significantly when using only flight data to predict hotel data. The major problem is the month of December, which is radically different from its

Author's Draft

preceding months. Because of this difference, the "error" carries forward to January, February and March, since we use a 3-month training window, and for these months, the training window includes December data.

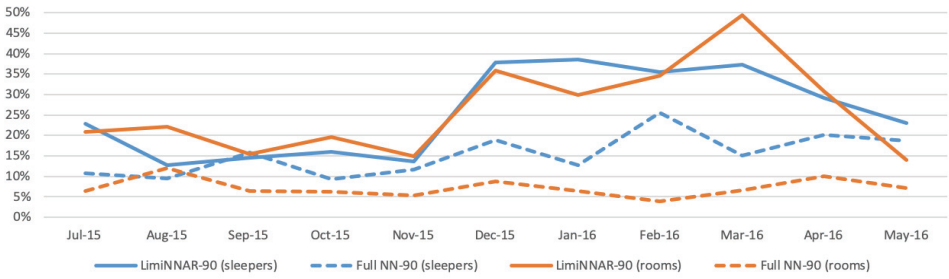


Figure 5.24 Prediction quality on full and limited feature sets

As can be expected, the nature of the problem at hand (i.e. tourism in Glasgow) is such that hotel business performance is intricately linked to the flights arriving into the city. Therefore, while flight data alone can partially predict the overall levels of hotel business, optimal predictive performance is attainable only when data for the levels of hotel business is readily available. While this may spell some trouble for our project, fortunately, not all is lost. In reality, the hotel datasets available to us contained a further bit of information which we so far discarded – this is the "forward bookings" at each reporting month. Along with the actual booking statistics for a given month, the company also provides advance booking information for upcoming months, so when reporting for March, the company shows not just the number of booked rooms, sleepers and revenue at the end of that month, but how many reservations have been made for the following April, May, June etc. (up to 12 months). Similarly, for Skyscanner, for a given month, we had been considering all flights arriving into Glasgow on that month, irrespectively of when that flight search was made. However, we can consider this dataset somewhat differently, and synthesize a similar metric of "forward bookings" by considering for each month, the number of flights arriving within that month but whose search (or redirect) date is previous to the 1st of that month. Therefore, while in our first round of experiments we considered an "optimistic" situation, assuming all necessary data was available on demand, and in our second round of experiments we considered a "pessimistic" situation, assuming that only the hotel or flight data was available on demand, we now explore a more "realistic" situation, in which an agency wants to produce business metrics forecasts for the upcoming month, on the 1st of that month, using data from the previous months (assuming they are all available at the end of the

Author Draft

month) and insights (forward data), hinting at what's likely to happen in the upcoming month.

Thus, for any given month for which we want to produce forecasts (target month), we take a training data set concerning a period of n months prior to the target month, and train a NN to forecast actual feature values from the "insights" that were available for the months in that period. We then use only the target month's insights to as input to the trained NN model, and assess the quality of the predictions (Figure 5.25).

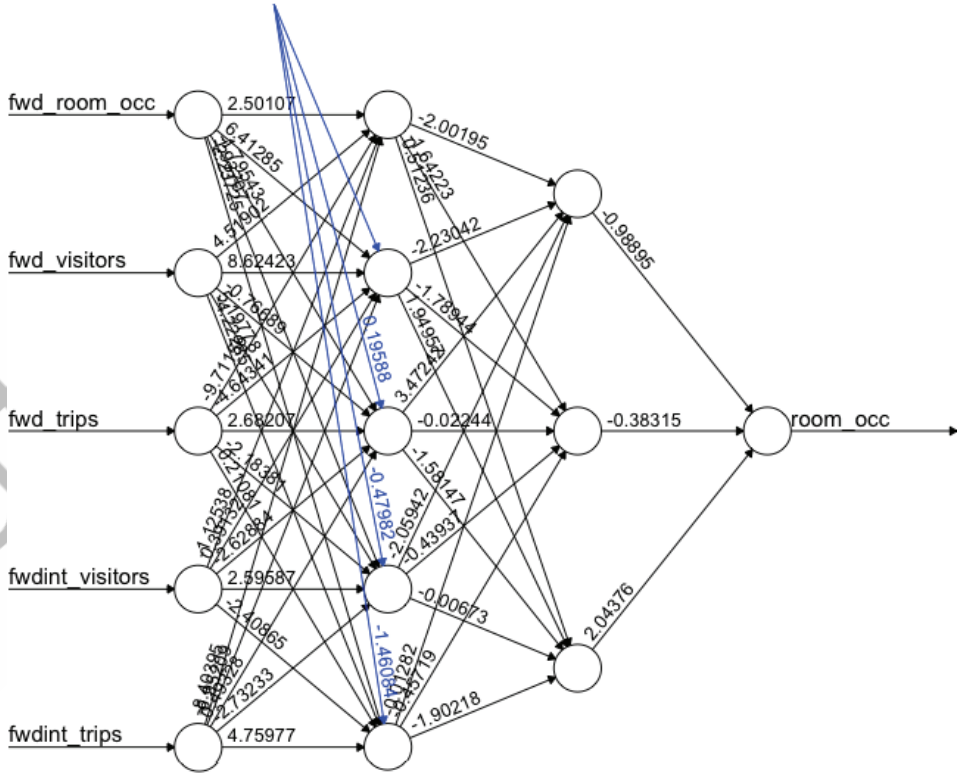


Figure 5.25 Learned network

As before, we use a 3-month training period, and converge the resulting models 30 times, reporting on the average NRMSE across all models. For this final type of analysis and for reasons of brevity, we present only the results for predicting the actual room occupancy and sleeper occupancy of hotels. The results are shown in Figure 5.26.

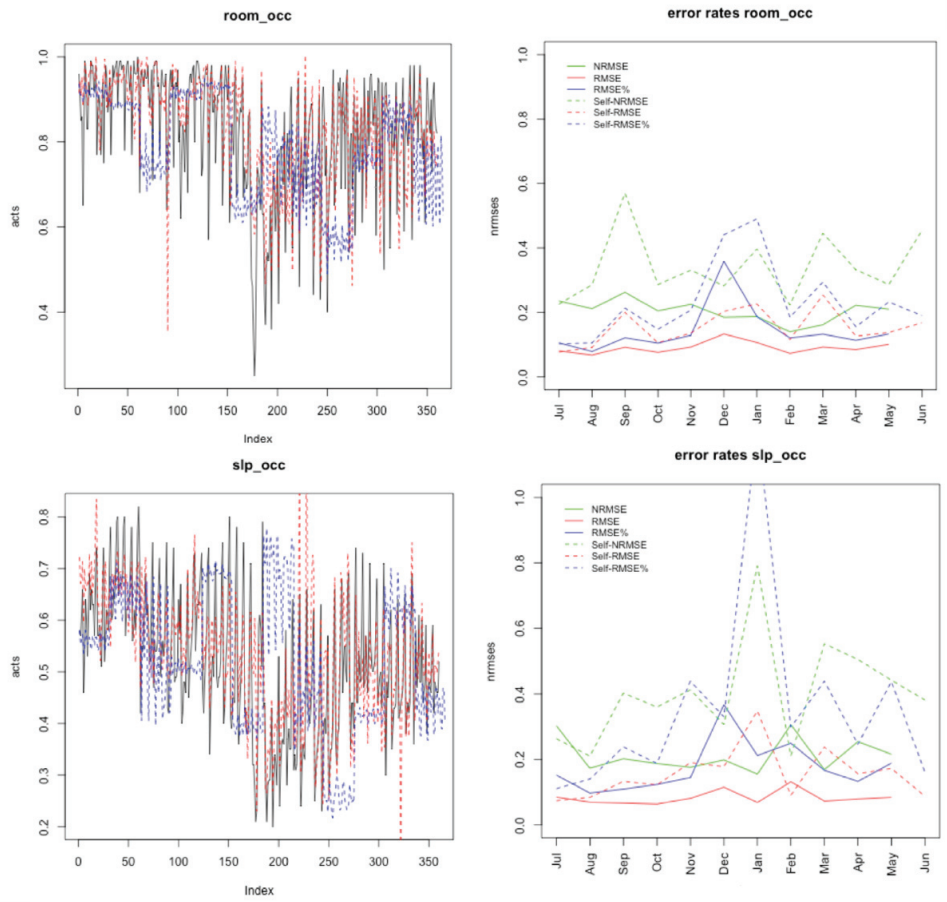


Figure 5.26 Prediction results and error rates using forward information as input features.

We note that the NRMSE hovers here around the 20% mark, a result that is comparable to the use of the limited dataset analysis (i.e. using only flight data to predict hotel occupancy).

To summarize the results so far, we have explored a range of methods to help us perform predictions on our dataset. Starting with the simple univariate auto-regression techniques, we moved on to a multivariate autoregression using a limited feature set (applying the PCA technique), and then performed a range of multivariate predictions using neural networks and varying the input feature set. For the latter, we explored the performance of neural networks assuming the full two datasets were

available in time for predicting the next month (optimistic case). We then attempted to predict features from a dataset, using inputs exclusively from the other dataset (pessimistic case). Finally, we explored the "middle ground", where we assumed that one of the datasets would not be available, but that we had "forward" insights into what that dataset might look like (realistic case).

Overall, each method shows what is theoretically achievable with these techniques (optimistic case), and how close we can get to this theoretical maximum under different real-world conditions, which affect the timely availability of data. An overview of these differences is shown in Figure 5.27, for the ability to predict our two key targets (room occupancy and sleeper occupancy). As can be seen, the absence of information has a clear impact on our ability to correctly predict. Overall, however, in the most realistic condition for our case (taking advantage of forward data), our NRMSE is around the 20% mark on a month-to-month basis, meaning that we are able to predict hotel occupancy figures within around 20% of what they actually are (i.e. if room occupancy actually turned out to be 80%, our prediction would most likely fall somewhere between 64 - 96%). This performance seems far from perfect, however, this figure is skewed by the normalization: when this error rate is computed as a simple RMSE, it is actually closer to the 10% mark and thus in reality, we are able to guess room and sleeper occupancy ±10% of what it actually turns out to be. Not bad at all!

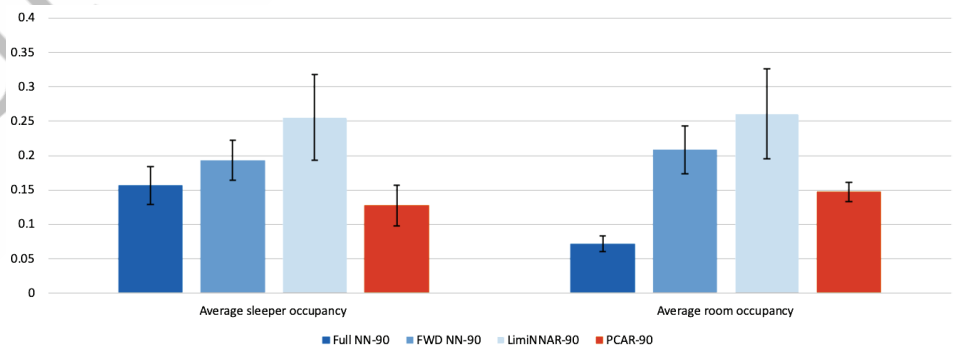


Figure 5.27 Overall comparison of the predictive performance attained under different conditions (NRMSE average for 07/15 - 06/13 with 3 months training)

5.5 Simple visualisation

In discussion with GCMB we developed a simple mobile visualization to help hoteliers plan services. Figure 5.5 left shows an absolute visualisation that shows on a blue-red scale how busy Glasgow is likely to be based on our modelling. This is a very simple visualisation of our modelling that can give a quick glance and also

Authors Draft

accounts for slight errors - our modelling will generally be no more than one colour step out. In discussion with GCMB though we discovered that to many in the hotel industry what matters most for bookings is whether the city is likely to be similar to the same day last year - busier or quieter. As well as staffing levels, staff bonuses are often based on beating last year's level, so nights that will be quieter a good target for discount pricing regardless of absolute level of busyness. Figure 5.28 right shows our variation highlighting quieter, roughly equal and busier nights.

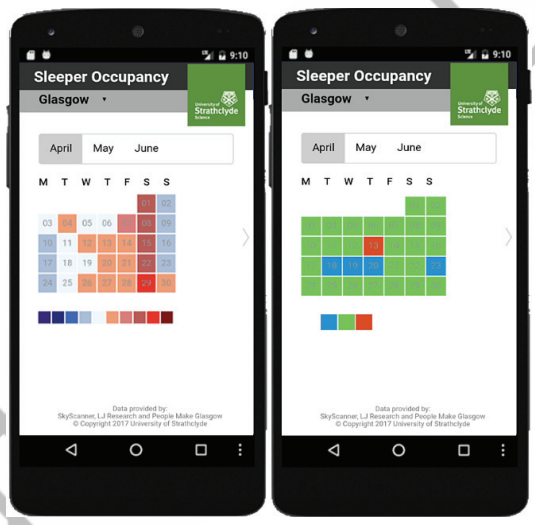


Figure 5.28 Absolute level of busyness and relative to last year simple mobile visualizations

5.6 Summary

Despite the lengthy investigation that we described in the previous sections, this case study is far from concluded - there are almost endless variations and tweaks to performance that could be applied to further improve our predictive performance. For example, we might:

- Tune the NN parameters to find optimal settings for our dataset
- Investigate performance with different size training sets
- Include more data sources into our models (e.g. weather information and special event information)
- Improve predictive performance for "difficult" months by training models for these months not on the data of preceding months, but data from equivalent periods in previous years
- Convert the problem from a regression task, to a classification task (i.e. assign categories to hotel occupancy, such as "very quiet:0-10%", "quiet: 11-20%",

Author Draft

… , “full:91-100%”), and try out different machine learning algorithms that are better suited for classification tasks.

However, from this project, we can extract some useful lessons that apply generally to any type of machine learning task, and which you may consider in your own practice. These lessons are:

- Understand your dataset: before attempting to go into creating complex ML models, you should take the time to get a “feel” for your data, exploring it in simple ways, and understanding where does the data come from, what does it look like, what does it mean, and why does it look like it does. As an example, in our case, we found that the humans who generated the flight data are not always reliable, either through intent (e.g. someone trying to book a flight for 500 people), or because their goals do not align to what we are trying to do. For example, most of our flight searches were for just 1 person, while flight redirects included a much larger number of cases where 2 or more people were being booked for. This indicated that there exist many use-cases where people do hasty flight searches to “generally check prices”, rather than to actually book, and do not bother inputting all the right information (after looking, we saw that the default number of passengers value on the Skyscanner website is 1). So, do not assume that you understand the data - sometimes, a fair bit of investigative work, even qualitative research, is necessary before you embark on a project. A close relationship with stakeholders, in our case primarily the Glasgow City Marketing Bureau, can help discover patterns that are obvious to them but not to researchers and modellers.
- Garbage in - garbage out: Huge datasets with hundreds of features are great, but not every feature is helpful. In fact, including more features than you need in a model can result not just in slower execution times, but also worse predictive performance overall. Take the time to consider which features are truly meaningful for your project - in our project the expectation that number of footfalls on the main shopping streets would be linked to busyness proved invalid as these counts are dominated by locals, not tourists. Also spend time considering how you might best transform your data, e.g. by grouping or averaging to reduce the number of cases, or by synthesizing new features from existing ones, which might prove more helpful (i.e. the day of the week, rather than the precise date). Finally, spend time to sanitize your data, especially when that data comes from humans.
- More is not always better: In our investigation, we saw that feeding lots of training data into the model actually reduced its performance. This is not

Author's Draft

unexpected, since our data exhibits a yearly seasonality. People's travel and tourism patterns for an area change from season to season, and thus feeding data from July into the model, with a target to predict December, results in worse performance. Take into consideration the nature of the dataset and the nature of the task at hand, and limit the exploration space by talking about the problem with domain experts - adopt their knowledge and consult with them, what they tell you is "about right", probably makes a lot of sense.

- Data is not always readily available: When it is, then you can build really good models with amazing performance. But the data you need isn't always going to be there on time (or at least, maybe not in the processed and sanitized form that you need it). Of course, go for the best-case scenario. But do not forget to also build and test models that do not assume perfect operating conditions, exploring bad and worst-case scenarios as well.

Acknowledgements

The project was funded by DataLab and was conducted in tight collaboration with "People Make Glasgow" - the Glasgow City Marketing Bureau. We are particularly grateful to Daniel MacIntyre of GCMB for his insights throughout the project. We are also grateful to Skyscanner and LJ Forecaster for their contributions to understanding their data sources. Finally we thank Strathclyde's City Observatory for their support in the project.

Author Contributions

Wilson (principal investigator) and Dunlop jointly gained the DataLab funding for the project and led most of the meetings with the data partners and funders. Komninos conducted the bulk of the modelling studies, analysis and visualisations. The chapter was jointly authored with Komninos as lead author.

References

Bangwayo-Skeete, P.F. and Skeete, R.W. 2015. Can Google data improve the forecasting performance of tourist arrivals? Mixed-data sampling approach. *Tourism Management 46*, 454–464.

Chu, F.-L. 2004. Forecasting tourism demand: a cubic polynomial approach. *Tourism Management 25*, 2, 209–218.

Dinis, M.G.F., Costa, C.M.M., and Pacheco, O.M.R. 2017. Similarities and correlation between resident tourist overnights and Google Trends information in Portugal and its tourism regions. *Tourism & Management Studies 13*, 3, 15–22.

Gretzel, U., Sigala, M., Xiang, Z., and Koo, C. 2015a. Smart tourism: foundations and

Author's Draft

developments. *Electronic Markets 25*, 3, 179–188.

Gretzel, U., Werthner, H., Koo, C., and Lamsfus, C. 2015b. Conceptual foundations for understanding smart tourism ecosystems. *Computers in human behavior 50*, 558–563.

Gunter, U. and Önder, I. 2016. Forecasting city arrivals with Google Analytics. *Annals Of Tourism Research 61*, 199–212.

Hyndman, R.J. and Athanasopoulos, G. 2018. *Forecasting: principles and practice (Second Edition)*. OTexts.

Li, X., Pan, B., Law, R., and Huang, X. 2017. Forecasting tourism demand with composite search index. *Tourism Management 59*, 57–66.

Padhi, S.S. and Pati, R.K. 2017. Quantifying potential tourist behavior in choice of destination using Google Trends. *Tourism Management Perspectives 24*, 34–47.

Rea, A. and Rea, W. 2016. How Many Components should be Retained from a Multivariate Time Series PCA? *arXiv [stat.ME]*. http://arxiv.org/abs/1610.03588.

Ruiz, A. 2019. The 80/20 data science dilemma. *The 80/20 data science dilemma*. https://www.infoworld.com/article/3228245/the-80-20-data-science-dilemma.html.

Scottish Enterprise and Glasgow City Marketing Bureau. 2014. *First to Know: Your Insight to Tourism & Events in Glasgow*. .

Xiang, Z. and Fesenmaier, D.R. 2017. Big Data Analytics, Tourism Design and Smart Tourism. In: Z. Xiang and D.R. Fesenmaier, eds., *Analytics in Smart Tourism Design: Concepts and Methods*. Springer International Publishing, Cham, 299–307.

Yang, X., Pan, B., Evans, J.A., and Lv, B. 2015. Forecasting Chinese tourist volume with search engine data. *Tourism Management 46*, 386–397.

Yang, Y., Pan, B., and Song, H. 2014. Predicting hotel demand using destination marketing organization's web traffic data. *Journal of Travel Research 53*, 4, 433–447.

Author's Draft