An informal summary of the work done at On The Beach Plc.

By Bilgin Sherifov

TASK/PROJECT: Collected data and developed econometric models for measuring impact of offline and online ad campaigns (dynamical systems modelling using Kalman filters, various regressions, and RNNs)

For this project I was asked if there is a way we could put a concrete number to what the impact is and what the returns are of our TV ads. This was the whole brief. The biggest challenge was that we had run TV campaigns only in the period starting with Christmas and ending by the end of March. This also coincides with the peak of our very strongly seasonal pattern of sales (in UK, people book beach holidays mostly in this period, with the peak being the first two week of January). The fact that the TV campaign so strongly correlates with the seasonal patterns makes it very difficult to disentangle the two by any means. The other challenge is the lack of price information about competitors (and this is a very price-sensitive business where customers are spoiled by alternatives). Another challenge is that that the effectiveness of a TV campaign depends not only on spend (which somehow should correlate with how many people saw it), but also by its content (not all TV ads running on a TV channel are equally effective). The most important challenge, though, is that TV ads impact not only the immediate (or near immediate) sales volumes, but they might have much more long-lasting and far reaching effects. For example, they might increase the brand reputation and awareness (and thus market share), which might reduce spend costs for our other major marketing item: keyword bidding on Google's and Bing's Pay--Per-Click (PPC) bidding. It is much more difficult to capture these long-term and indirect impacts.

So far, in phase one we have focused only on the direct impact of TV ads. One of the modelling ideas I came up with was to try to build a model that would explain as good as possible our booking volumes without using TV ad data, with the hope that it will predict lower estimates for the periods when TV ads were running (and in the subsequent few months, as well). So, this is a counterfactual analysis (in a way similar to pre- and post-treatment studies in medicine, but with much more granular data). The other idea was to include spend for offline ads as one of the inputs and then use model outputs with and without offline spend to determine the incremental value that TV ads add. These two models were further split into versions with constant parameters and versions with time-varying parameters (to account for the fact that over the years the business and the market might undergo significant changes, thus factors that were once important could in time become less important and vice versa. A good example is the GBP/EURO exchange rate, which was probably of little concern when the Pound was strong but has become a major factor in recent years). Another take on the time-invariant models was boosted using regression trees, like LightGBM and XGBoost.

Most of the time spent on this work has been about collecting the right set of data. Currently, after trying and discarding many others, we are using these inputs:
- Number of people employed (as a measure of market size)
- Consumer Confidence Index (as a measure of market mood)
- GPB/EURO exchange rate (as a measure of how expensive/cheap the product s are)
- The average position we have attained in PPC bidding (as a measure of ur visibility in search results)
- The average cost per attained position on PPC bidding (as a measure of how aggressive the competitors were in those bidding)
- The margins we put to our products (as a measure of how competitive our prices are with respect to competitors')
- How much competitors have spent on TV ads (as a measure of their aggressiveness in the TV ad space).
- Season (month or week of the year)

All inputs were augmented with their time delayed versions (up to 4 time steps when the data was aggregated into monthly granularity), as well, thus ensuring we can capture the dynamics of a linear mode via impulse-response modelling.
We split bookings data into thirteen TV regions, two visitor types (new and returning) and four channels through which the customers had reached us on the day of purchase. The models were fit to all these 104 time-series (each spanning nine years).

The models themselves are in terms of multiplicative impacts each input has on a constant base value (say employment for May 2017 has increased the base value by 20% while GBP/EUR has brought it down by 10% and similarly for all inputs). This was linearized by taking the logarithm of the bookings. This turns the model into a linear multivariate regression. To put some business constrains and prevent the optimization algorithms from coming with good fitting, yet meaningless estimates, I changed the inputs so that an increase in one would never imply a decrease in the outputs. Then, for the constant-parameter model I used non-negative least square optimization to fit the models.

For the time-varying model, I defined the parameter evolution in a state-space format, with identity transition matrix, driven by white noise and time-varying observation matrix composed of the input values. Again, I imposed non-negativity on the parameters. Then I used the Kalman filter to fit the parameters and the the Rauch–Tung–Striebel smoother acting on its results to smooth them out.

I've also tried state-augmentation with a hidden state variable to try to refine the mode in such way that the base value is not a constant but evolves with time.

I'm also working on implementing the same ideas as recurrent neural networks (rNN) and use PyTorch's Autograd to fit even more complex models. This time I hope to capture also interactions between various channels (brand channels suppressing non-brand channels etc.) and the long-term impact brand strength has on improving market share. A pilot study and a first implementation of these ideas was presented at MancML talk in mid- March 2019 and was received with great interest by some in the audience with whom we might collaborate to push the ideas further.

One of the additional ideas here is to use LSTM units that might help capture long range interactions (again, there is an early implementation of this).

I have also tried VARIMAX type of modelling.

All work was done in Python.

TASK/PROJECT: Improving algorithms for product placement in on-site search results (recommendation systems)

Once visitors conduct a search for a given destination, a vacation period, and passenger composition, we have to present them a list of hotels. We have to decide how to order the hotels on this list, before customers start interacting with various filters. For this we use various metrics that we believe are important for the customers and for the company. This is done differently for visitors with prior history on our site and for first time visitors. So far, I've improved only the later one (recommendation system for new visitors) and I'm currently working on improving the first one (recommendation/personalisation system for visitors with prior history). I've fine-tuned the old algorithm to correct for certain anomalies in the orderings and came up with a way of weighing how much each metric should impact the final score.

This is an automated process and is all done in Python.

TASK/PROJECT: Exploring whether there is a basis for extending the work described above, but segmenting not only by destination and passenger composition, but also by departure airport and season

I used hundreds of millions of search data to see if there are patterns that would justify a segmentation based on departure region and season. I used almost all of the clustering algorithms available in Python's scikit-learn package and tSNE and Umap manifold projection algorithms for visualization purposes. I also used a Javascript library (d3.js) and the force graph module there as a means for both interactive visualization and for segmentation (it uses actual physical model of charged connected particles to achieve good visualization in 2D, which also results in good segmentation).

All work was done in Python and some in Javascript and HTML.

TASK/PROJECT: Worked on improving company's bids for keywords in Google Pay-Per-Click (PPC) searches

The bid-optimization code written by my predecessor was a huge organically grown Matlab code. My very first job on this role was to go through the code and reverse engineer it. At the end of this work, I proposed that we rewrite the tool in Python and simplify it.

Part of the tool is assessing how much each digital ad channel and each key word has contributed to the overall profits (called the "attribution" problem) and is using this assessment to decide a new bidding value. I argued that this assessment method needs to be changed and that instead of relying on purely data-driven approach, we should include expert views of people with deep domain knowledge and build a hybrid system.

We had tried other data driven approaches such as Shapley value (previously used by Google but later they switched to Rubin causal model and finally scrapped the whole attribution tool) and survival analysis. My core argument has been that all models which try to do partial credit assignment are not quite appropriate for this problem since they either impose (distorted) linearity on the underlying inputs or are optimal for problems where one has to share fairly something of a fixed quantity to all actors (keywords, channels etc) and this doesn't apply to the problem at hand. In the process, I had also built various simulators to get a better understanding of the data and the tools used in the Matlab code.

All work outside of Matlab was done in Python.

TASK/PROJECT: Worked on improving the tools for split-test analysis

This was the first project to which one of the new junior members of the team was assigned to. My role was to guide him, help him with ideas, and double check his work. He built the tool in R. We went beyond using standard statistical tests and instead we also looked at the evolution of the test statistic and test results. We also built tools to test whether the data distribution is the appropriate one for the tests used. Finally, there was an attempt to do Bayesian analysis via Monte Carlo simulations.

All work was done in R, though I didn't write any of it.

TASK/PROJECT: Analysed user journeys and behaviour and currently building models to predict propensity to purchase

This is an ongoing project. At the current stage I have focused on trying to predict the propensity to purchase based on only data from the current session of a visitor (ignoring his/her prior visits at this stage). Most of the work so far has been about collecting the data, transforming it and preparing it for the models that will be used (initially Naive Bayes, Logistic Regression, neural network classifiers etc). For example, Naive Bayes works best with uncorrelated features. However, many of the features in the data are correlated. So, some sort of feature whitening should help (like combining device, operating system, and browser into a

single feature). Also, some of the categorical features have tens or hundreds of values, but with only a few of them being observed significantly often.
I've also written a code to visualize individual user journeys on our side and use these to gain insights and come up with modelling ideas.

All work done in Python.

TASK/PROJECT: Image labelling
Built a pilot study, as proof of concept, that can successfully label almost all of the hotel-related images on our site. This is to be but into production soon and its outputs will be used by various departments for their own needs.

All work done in Python.

TASK/PROJECT: Built automatic visualizations for various marketing-related measures
On a daily and hourly basis, we generate various graphs about our bidding performance and share these with certain stakeholder in the marketing department.

This is an automated process and is all done in Python.

TASK/PROJECT: Built simulations of various processes to be analysed and modelled
For almost every project I have worked on, I try to build simulations of the processes that might be generating the data (ensuring that the data generated via the simulations has similar distributions to the actually observed data). Then I use these simulated data to test various models and estimation procedures, to understand what the models and the identification algorithms can capture and what they cannot (in a way, these are my controlled experiments). Only then I apply selected models and algorithms to the actual data.

TASK/PROJECT: Recruited, trained, and lead junior data scientists
During my time here, there were two rounds of recruitment: the first time for junior data scientists and the second time for a mid-level data scientist. For both rounds, I prepared the tasks that we sent to the candidates for them do at home (classification, clustering, and regression tasks with some visualization for the junior candidates and estimation of dynamical systems models for the mid-level candidates). Also, I usually run the technical parts of the face-to-face interviews, in which I ask the candidate to walk me through his/her code, explain what and why he/she is doing, and try to understand not only the depth of his/her knowledge but also the way of his/her thinking (we'd go in so much detail that some interviews have lasted for more than two hours).