Project: Evaluating user abandonment due to page performance

Bayesian Statistics - Fall 2021 ISYE6420

StudentID: oansari3 (Omer Ansari)

Date: 12/5/2021

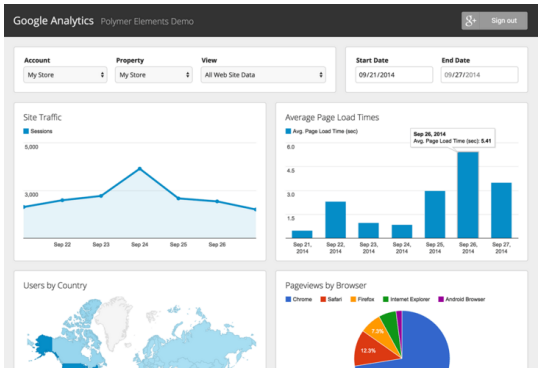
Summary

Studies have shown that poor web page performance causes consumers to lose interest and bounce (abandon their session). In this project, I analyzed data from a public website for the month of October 2021 to accomplish the following goals using Bayesian Statistical Methods

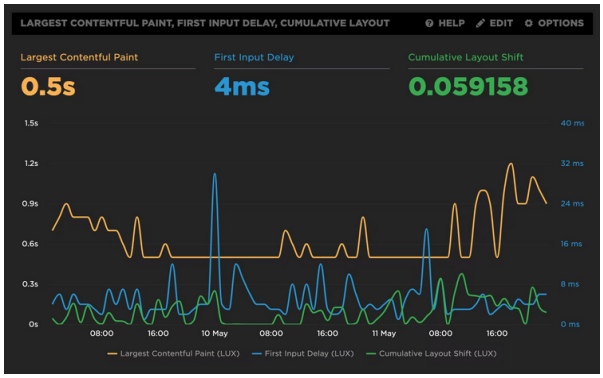
1. To identify, out of all page latency variables available, which single predictor explains the page abandonment the most
2. To assess the Bayesian model's prediction accuracy the level of page abandonment using the single predictor.

Data Collection & Transformation

Two sources were used to collect data. These were Google Analytics (GA), and SpeedCurve. Both these systems are software as a service (SaaS) applications which are connected with the website in question. GA uses a push mechanism to collect data: When a user interacts with the website, information from that user's browser is sent to GA. Speedcurve's method of data collection is a pull method. It polls the website, acting like a browser, several times a day and measures and aggregates the page latency statistics.



Google Analytics



Speed Curve

The following specific data for the website was extracted from these systems:

- Document Content Loaded Time : The **average time (in seconds)** that the browser takes to parse the document and execute deferred and parser-inserted scripts (DOMContentLoaded), including the network time from the user's location to your server.
- Page Load Time : The average amount of time it takes for a page to show up on your screen. It's calculated from initiation (when you click on a page link or type in a Web address) to completion (when the page is fully loaded in the browser)
- Speed Index: The **page load performance metric** that shows you how quickly the contents of a page are visibly populated. It is the average time at which visible parts of the page are displayed.
- Time To Interactive: How long it takes a page to become *fully* interactive. A page is considered fully interactive when the page displays useful content, the page responds to user interactions within 50 milliseconds
- First Contentful Paint : **when the browser renders the first bit of content from the DOM**, providing the first feedback to the user that the page is actually loading. ... The First Contentful Paint time stamp is when the browser first rendered any text, image (including background images), non-white canvas or SVG.

As you can imagine, each of the following statistics can have an influence on the user's psychology, either keeping her on the page or triggering her to leave the website. This is why I pulled all these metrics (as daily averages) for the website.

To measure the page abandonment, I used the bounce metric:

A *bounce* is a single-page session on your site. In Analytics, a bounce is calculated specifically as a session that triggers only a single request to the Analytics server, such as when a user opens a single page on your site and then exits without triggering any other requests to the Analytics server during that session.

This is the closest metric that can be used to measure a user departing a page, and is better than the exit metric, where some modicum of interaction may have already happened with the

page. Also, every user eventually exits a page and she decides to stop browsing, and the cumulative exit rates are always 100%

Data was pulled from these two systems, aggregated and visualized within Excel (References) and a consolidated csv file was extracted for further analysis (References)

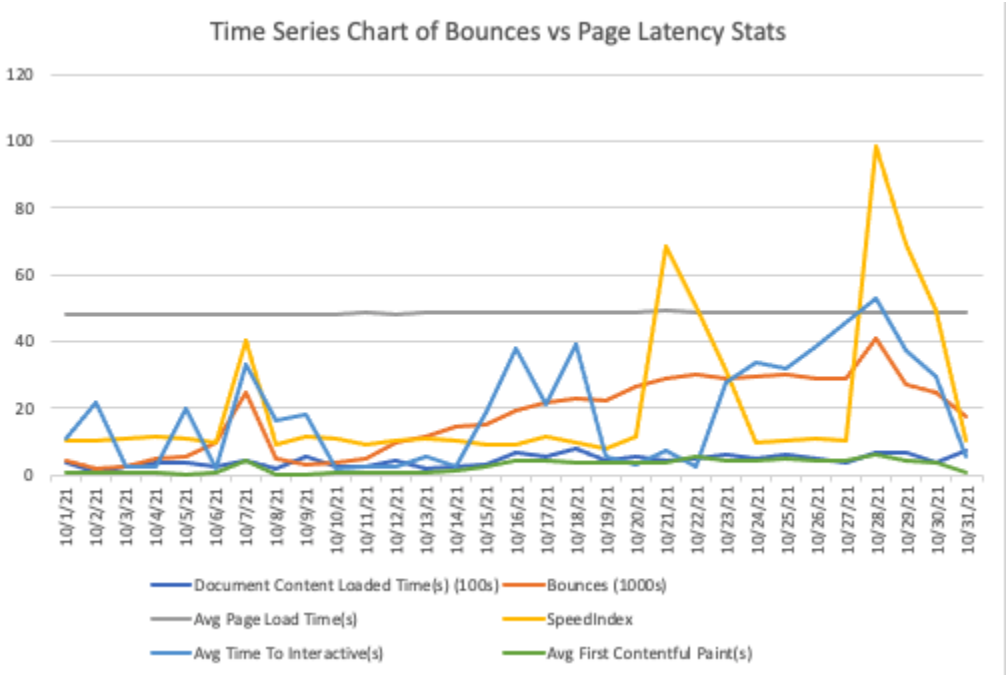
Data Analysis

Analysis was done using an Openbugs script (References) as well as Jupyter (Python) Notebooks (References). For the notebook, Pymc3 was used as the Bayesian library to accomplish the above stated goals.

Let's revisit the first goal:

Goal1: To identify, out of all page latency variables available, which single predictor explains the page abandonment the most

First, let's take a look at the data in a time-series format. Below, the bounces have been and cumulative document content loaded times have been scaled down, so all the various graphs can be seen together.



As we can see, there are some page metrics which seem to be correlated positively with the number of daily bounces.

To accomplish the goal of specifically identifying the most informative predictor, I used Openbugs (see project.odc). I commented out the various linear regression equations and ran the script with 1M samples, and 100k steps. (The multivariate model took 16 seconds to run). I was able to derive the following R squared values for each linear regression equation. The relevant snippet of Openbugs code for the multivariate model is as follows:

```
bounces[i] ~ dnorm(mu[i], tau)
mu[i] <- b0 + b1*doc_load_time_cumulative[i] + b2*avg_first_contentful_paint[i] +
b3*avg_time_to_interactive[i] + b4*speed_index[i] + b5*avg_page_load_time[i]
```

The bounces are assumed to follow a normal prior distribution, with the mean of this distribution influenced by all the various variables in a linear regression model.

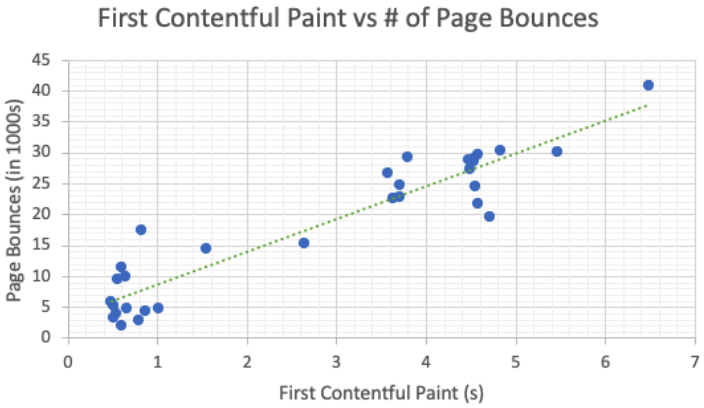
Variable	R squared
all	89.34%
Cumulative Document Load Time (s)	38.81%
Average First Contentful page (s)	**87.01%**
Average Time to Interactive (s)	30.59%
Average Speed Index	29.29%
Average Page Load Time (s)	-0.0006%

The most informative predictor for users bouncing off the page is FCP.

First off, the high R squared score for the full model is a little surprising. Page performance is a contributor but not the sole contributor of why people bounce off a page. Another reason is stale content (the returning user looks at the content, and seeing that it's stale, bounces), or inaccurate targeted advertising (i.e. you advertise it to the wrong audience, they click on the link, get to the content, decide it's not for them and bounce). In this case, given the score is so high, we can assume the other reasons are neutralized, and page performance indeed is primarily driving people’s behavior.

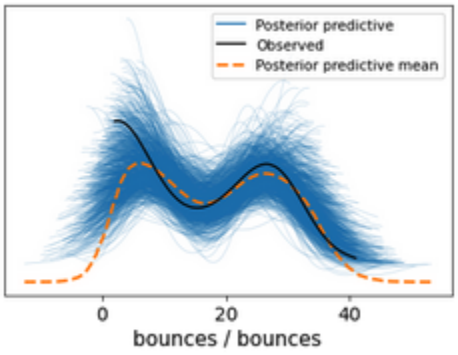
Second, we got a very small (even -ve) R squared for the average page load time. A negative R squared signifies a worse fit than a horizontal line. This actually makes sense. The website has a lot of right graphics and outbound links which take a while to fetch, render, display. In several cases, there is really nothing to display either; the javascript on the page is executing async

code (such as sending data to Google Analytics) which the user is not bothered with, nor does it block the user.



Plotting the data of the first contentful paint against the daily bounce, the graph is a linear relationship. This makes logical sense. When the document load time sum of ALL the pages the viewers viewed that day was low, there was much less bounce. As the load time increases, the slope of the bounces goes higher. Note that it should not extend to infinity, given there is only a finite audience coming to the website. Even with the worst case of 100% bounces, that would still flatten out. That is not reflected in this graph due to the limited number of data for high FCP.

Goal2: To assess the Bayesian model's prediction accuracy the level of page abandonment using the single predictor.



I used bambi (Arxiv:2012.10754) in a google colab (Jupyter) notebook to create the posterior of the linear model. The blue lines represent the posterior predictive distribution estimates, and the black line represents the observed data (which is # of bounces/day in 1000s). Our posterior predictions seem to perform an adequately good job representing the observed data in all regions except closer to 0, where the observed data and posterior estimates diverge. This can be explained as follows. Low FCP values necessarily do not mean we will have lower # of bounces. The average is of the actual # of bounces is higher than the mean, which implies the users were bouncing off the page not because page performance issues (obviously, since FCP was low), but because of other intrinsic reasons (stale content, or inaccurate social advertising)

References

- Openbugs odc script for Goal1 (link)
- Google Colab notebook Goal2 (link)
- Microsoft Excel sheet where data was aggregated and cleaned (link)
- Curated csv data (link)