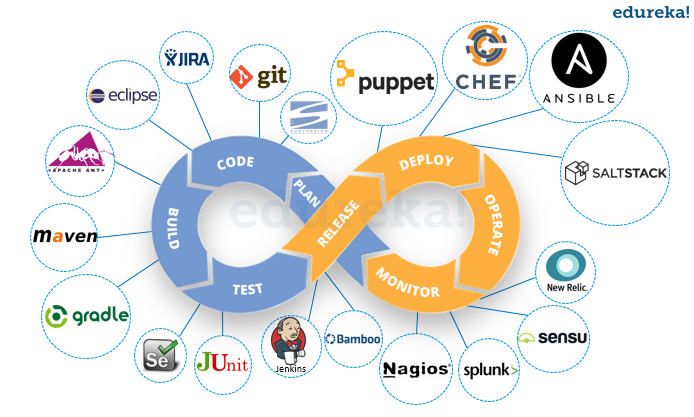
**Understanding ML DevOps**

To start Naïve, CI stands for Continuous Integration and CD stands for Continuous Delivery/Continuous Deployment. You can think of it as a process similar to a software development lifecycle.

DevOps is very self-explanatory if one looks at below mentioned chart.

[Credits]:Edureka



Let’s take example of the application we are building for Marketing leads prediction which is going to be maintained using above displayed devOps pipeline, with just a little tweaking to convert to ML devOps.

First part of this Mobius strip starts incepts at planning and approvals.

I planned by first looking at the problem statement, followed by analyzing the data.

Data is the most important part of any ML application.

I did the data analyzing using python’s visualization libraries.

More into these in README.md file.

As I am alone building these, so I need to start building the logic to achieve the problem statement in least optimized possible way.

If group of developers are responsible for writing the code, who will further go on and build the web application. Now, when this code is committed into a version control system (such as git, svn) by the team of developers.

I did that using Git and pushed my source code to GitHub.

<https://github.com/jaytimbadia/mrkt_leads_conversion>

Next, it goes through the **build phase**, which is the first phase of the pipeline, where developers put in their code and then again, the code goes to the version control system with a proper version tag.

These is the part where code optimization occurs along with version controlling.

**My code is very simple, since my focus is to explore entire ML Devops pipeline instead of building model production ready.**

So we don’t require creating different branches and pull and merge operations.

Once we or any developer build the application part, that part needs to be tested before releasing it to the main branch for deployment.

So any developer working on any branch needs to make sure his code is working when integrated with entire application.

To do that, he/she performs various test like unit, system, integration, penetration etc.

Here, as we are alone, I just need to perform one single test to assure my application is up and running.

Once tested, I moved it to commit my all source code to GitHub.

We can automate build using **Jenkins**, which any developer can use to test their application for mater merging.

As we are working on Python, we don’t need to compile our code.

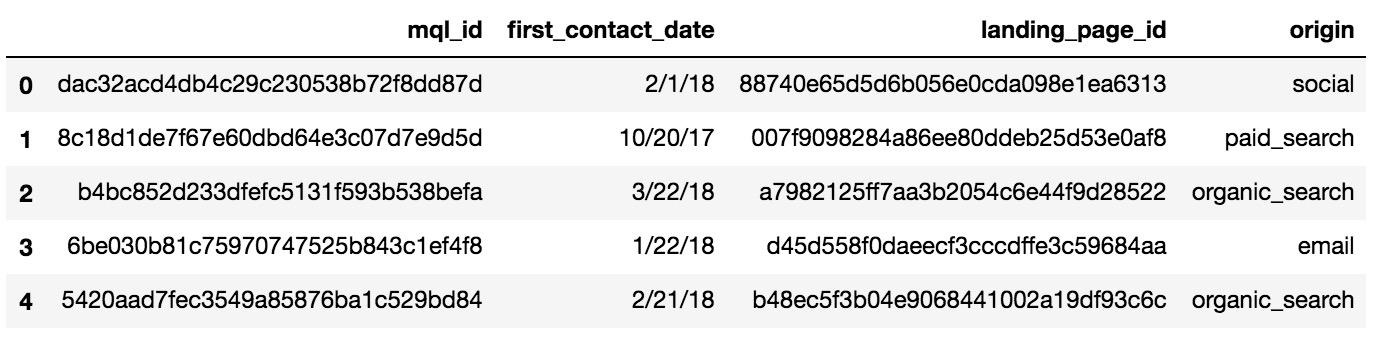
Once build is successful in Jenkins, we can deploy our application using docker or AWS or client network.

Later we need to operate, maintain & monitor our application using tools like Kubernetes, Ansible or Splunk for monitoring purpose.

**Dataset characteristics**

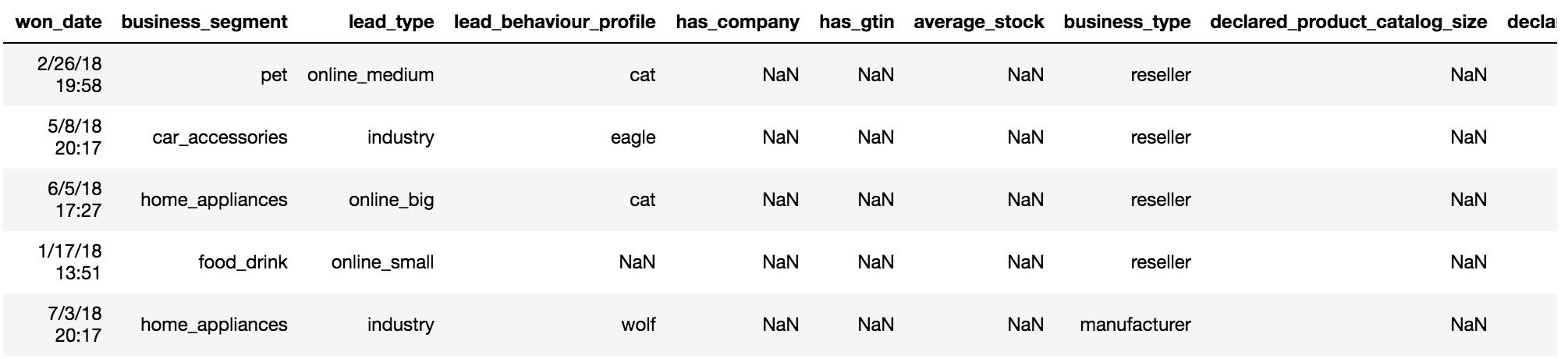
The datasets were obtained from Kaggle.

* First dataset contains 8000 data points about the MQL (marketing qualified leads), its first contact date, the landing page that captured the leads, and the lead origin (the channel that bring the leads to the landing pages). It has 4 variables with 8000 data points



* Second dataset contains 842 observations. Each observation is a won deal of Olist which consists of the mql\_id of the merchant, seller\_id (used in Olist platform), sdr\_id (the sales development rep that was in charge of the sales process) won\_date, its business segment, lead\_type, lead\_behaviour\_profile, business\_type, declared\_monthly\_revenue,... (It has 10 variables)

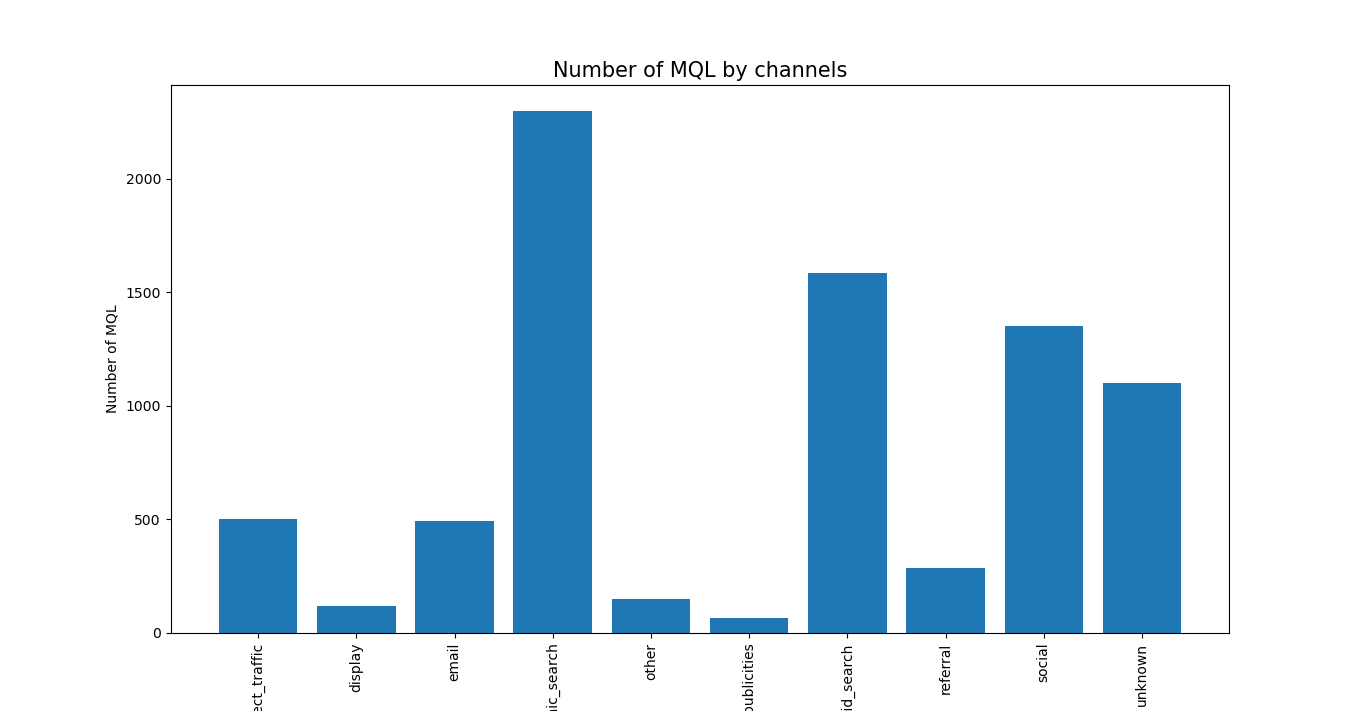


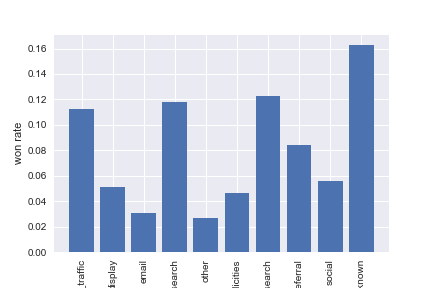


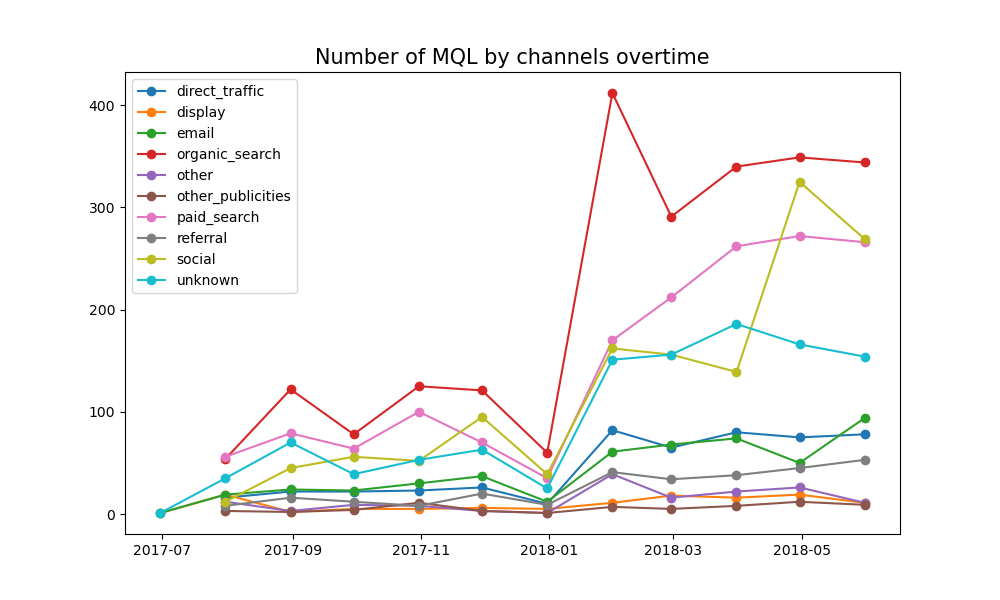
* Third dataset: Via Kaggle, Olist also donated their dataset about the demand side. We will use this to inform Olist marketers about the performance of the sellers, thus help them to improve the B2B marketing process.

**Exploratory Data Analysis**

**Channels:** The majority of MQLs come from organic\_search channel, followed by direct traffic and social . Other, other\_publicities, referral and display are the sources that bring the least MQLs to Olist. The organic\_search MQL is significantly increased in 2018-02 and declined after that. This can be the result of a big event/PR campaign.







**Landing page – Self Explanatory.**

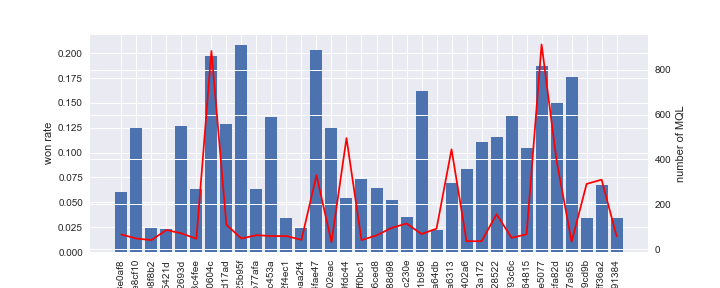


figure 4:

**Conclusion & Recommendation**

* We can look at landing page graph, from which we can figure out which page has much won or conversion rate against the number if leads hitting.
* We can also collect geographical data for further improvement in targeting leads.
* Here, we have use ML for predicting lead conversion based on landing page & origin information.