1. ­­­­The data set is in the JSON format. In order to validate the data:
   1. The file’s schema will be validated
   2. The data type will be validated
   3. IPs should be in the valid range
   4. Timestamps should be in the valid range
   5. The distribution of numerical columns will be checked and compared with historical data
   6. The frequency of the categorical columns will be checked
2. The data is loaded as a pyspark data frame. The JSON file has a nested schema in the form of:

root

|-- request: struct (nullable = true)

| |-- documentReferer: string (nullable = true)

| |-- error: string (nullable = true)

| |-- ...

| |-- frameId: long (nullable = true)

| |-- requestHeaders: struct (nullable = true)

| | |-- Accept: string (nullable = true)

| | |-- Accept-Encoding: string (nullable = true)

| | |-- ...

| | |-- x-ijt: string (nullable = true)

| |-- requestId: string (nullable = true)

| |-- requestType: string (nullable = true)

| |-- responseHeaders: struct (nullable = true)

| | |-- $WSEP: string (nullable = true)

| | |-- 'X-Propose-Redirect: string (nullable = true)

| | |-- - X-Frame-Options: string (nullable = true)

|-- server\_request: struct (nullable = true)

| |-- accept\_language: string (nullable = true)

| |-- country\_code: string (nullable = true)

| |-- ...

| |-- user\_map: struct (nullable = true)

| | |-- puma: string (nullable = true)

| | |-- tiger: string (nullable = true)

| |-- x\_forwarded\_for: string (nullable = true)

In order to make the data representative, useful columns are extracted using spark sql query. This will flatten the schema and gives us a pyspark data frame with correct format that is easy to work with.

Additionally, columns with time data are transformed from unix time to timestamp format (to make them more readable and extract more data from). With timestamp format we can extract year, month, hour of day, day of week, day of month and day of year.

Columns the hostname of columns consisting urls are extracted using regex.

1. Multiple data products can be built to show the company’s performance
   1. The interaction of the customer with the website, along with what they are doing before and after. (e.g. customer opening a clothing company website and making a purchase vs going to another clothing website without making a purchase)
   2. Customer experience; time spending on the website, number of pages/products visited, etc.
   3. The following parameters can also provide useful insight:
      * + Number of distinct days website visited by single customer
        + Total number of pages visited in a fixed amount of time
        + Total number of visits in a fixed amount of time
        + Most visited pages
        + Number of pages viewed per visit
        + Average views per page
   4. Search patterns can show attentions and intends (e.g. the correlation between number of google search of bitcoin and drastic increase of the price)
   5. The performance of the company among different age, gender, income, demographic, etc. groups can be estimated by analyzing the clickstream data
   6. Conversion ratio for an advertisement company/campaigns
   7. Number of visitors and amount of time they spent on the website
   8. Number of visits resulted in to purchase of products (both in terms of quantity and value)
   9. The visitors journey and bottlenecks on the website
   10. Number of websites that are referred form the company’s website
2. Using the clickstream data we can identify and evaluate the performance drivers of the company. For example:
   1. Compare and contrast customer behavior after engaging with different marketing strategies
   2. The effect of different types of advertisement
   3. The effects of the different website version on the conversion rate
   4. By analyzing customer trends, we can see the customer path to the company’s website
   5. We can find certain similarities and trends that identify the performance drivers
   6. In case of registered users, the demographic and marketing data can be employed to find the drivers too.
   7. Not only performance drivers can be found but also performance hinders can be found (e.g. top pages where most visitors dropped their journey) and improvements can be made
3. Couple of points can be identified about the user profile of domains of Amazon, Netflix and Priceline:
   1. The table below shows the percent share of each operating system across all three websites:

|  |  |  |  |
| --- | --- | --- | --- |
| OS | Amazon | Netflix | Priceline |
| Windows | 65 | 51 | 77 |
| Mac | 29 | 24 | 22 |
| Linux | 5 | 10 | 0 |
| Unknown | 1 | 15 | 1 |

Windows and Mac has the highest share between all websites, respectively. The interesting trend is the high share of Linux and unknown (mobile?) operating systems for Netflix which can be related to more tech savvy and young users.

* 1. Table below shows the number of visitors in percent for each day of week:

|  |  |  |  |
| --- | --- | --- | --- |
| Day | Amazon | Netflix | Priceline |
| Sat | 20 | 19 | 41 |
| Sun | 19 | 22 | 23 |
| Thu | 19 | 17 | 14 |
| Mon | 17 | 19 | 5 |
| Tue | 13 | 11 | 11 |
| Fri | 12 | 12 | 6 |
| Wed | 0 | 0 | 0 |

The highest number of visitors is for Sundays and Saturdays (weekends) across all the websites. Priceline experience a more dramatic reduction in number of visitors during the weekdays. Wednesdays has the lowest number of visitors across all three website.

1. The clickstream data set can be used for other purposes such as:
   1. Personalization: identify the best journey and website design for similar customers
   2. Fraud detection: identify the fraud cases both in advertisement and transactions
   3. Similar customers: customers with similar patterns, trends and demographic can be identified. The insight can be used in marketing, personalization etc.
   4. Next move: using Markov Chain models, we can train probabilistic model to predict the behavior of the customer.
2. To load and analyze the data Python and Pyspark have been used. Python is chosen because it is user friendly and powerful. Large selection of open source libraries is also available for python. One of the libraries that is used here is Pyspark which makes it possible to analyze big data in a distributed fashion. Pyspark package also contains distributed and efficient implementation of many machine learning algorithms which let data scientists build and train models on big data sets with billions of rows and thousands of features. One set back of Pyspark is that the overhead for small data sets make it slower than non-distributed tools such as pandas and scikit-learn. Also, setting running parameters such as executor-memory, executor-instances, executer-cores, etc. is not trivial and they effect the efficiency significantly.