Name: Deenu Khan

Email address: dr.deenukhan001@gmail.com

Contact number: 8800128247

Anydesk address: 784 174 127

Years of Work Experience: 2

Date: 18th Nov 2020

Self Case Study -1: **Airbnb New User Bookings**

Overview

Airbnb provides services like **homestays, tourism, lodging,** and an interesting fact is, places it provides during its services it does not own any of them, it simply works as a broker and receives commissions from each of the bookings. Airbnb is based in San Francisco, California, United States.

Now, when we have a little understanding about Airbnb, let's discuss what Airbnb wants to improve in its ecosystem or simply the business problem Airbnb wants to solve, **Airbnb wants to share more personalized content with their community or users, decrease the average time for a new user to do its first booking, and better forecast demand** and to do so, they want to know where (in which country) a new user will book their first travel experience with the help of Machine Learning and cutting edge Predictive Techniques.
A few of the advantages of such information beforehand could be, Airbnb can better utilize the resources and save money, and when Airbnb will provide better forecasts which is nothing better recommendations to its users then sales and visits on the Airbnb websites would be more likely to be increased which will surely result in more profit.

To achieve these tasks Airbnb has hosted a hiring contest on Kaggle in which they have given a dataset of USA citizens which includes their demographics locations, web session records, and some summary statistics. And you are asked to **predict 5 most likely countries in which a new user will book their first booking destination and rank them on the basis of likelihood in decreasing order**. There are total of 12 possible predictions a user can make, **'US', 'FR', 'CA', 'GB', 'ES', 'IT', 'PT', 'NL',**

'DE', 'AU', 'NDF' (no destination found), and 'other' (booking made, but country is not available in given list)

Their Evaluation Metric for this competition would be NDCG (Normalized discounted cumulative gain) @k where k=5. The details of the NDCG calculation are available **here.** and I will also be explaining this in a later section.

Dataset Analysis

Data for this competition is given in **6 different files, 4 of them are actually for training our Predictive Model and 1 is for Testing the model** and the other 1 is just a sample of submission format in which we need to submit on Kaggle. Data for this competition can be accessed from **here**:

One more important thing to keep in mind is that **training and test datasets are split by dates**. In the test set, we have to predict all new users which were first active after **7/1/2014 (MM-DD-YYYY)**, and in the sessions datafile records are starting from **1/1/2015 (MM-DD-YYYY)** while in the training data users are starting from the 2010 year. And the above information is telling us that **we don't have session information for all the training users**, so we need to take care of this aspect as well.

Now, let's deep dive into our 6 datafiles and understand what these correspond to, and find out what each feature is. Well, most of the features are self-explanatory but still have a look below.

1. **train_users.csv:** this CSV file contains the data for training our predictive model and it is having **213451 records**, it has **1 dependent feature or target variable and 16 independent feature** which are explained below, also we can see some basic information in below screenshots, like what are the **fields with NaN values, total_number of data points,** type of fields etc, here in train_users we we have 213451 total entries and **3 fields are with NaN values.**

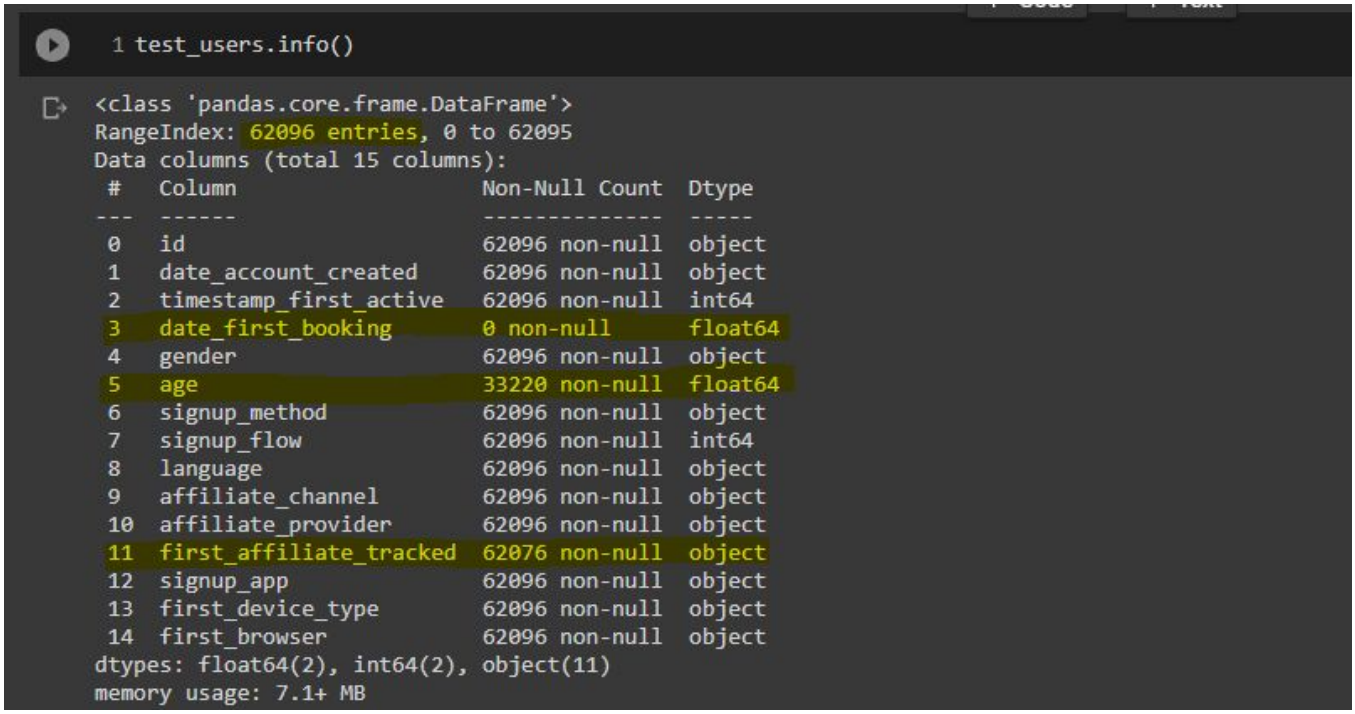
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 213451 entries, 0 to 213450
Data columns (total 16 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   id                       213451 non-null  object
 1   date_account_created     213451 non-null  object
 2   timestamp_first_active   213451 non-null  int64
 3   date_first_booking       88908 non-null   object
 4   gender                   213451 non-null  object
 5   age                      125461 non-null  float64
 6   signup_method            213451 non-null  object
 7   signup_flow              213451 non-null  int64
 8   language                 213451 non-null  object
 9   affiliate_channel        213451 non-null  object
 10  affiliate_provider       213451 non-null  object
 11  first_affiliate_tracked  207386 non-null  object
 12  signup_app               213451 non-null  object
 13  first_device_type        213451 non-null  object
 14  first_browser            213451 non-null  object
 15  country_destination      213451 non-null  object
dtypes: float64(1), int64(2), object(13)
memory usage: 26.1+ MB
```

a. **Id:** This is simply an alphanumeric user-id that is unique for each user.
b. **date_account_created:** This feature corresponds to the date on which a user has created his/her account, and this is **YYYY-MM-DD** format.
c. **time_stamp_first_active:** Timestamp **(YYYYMMDDHHMMSS)** when the user was the first time active on Airbnb Website**,** and **keep in mind this can be earlier than date_account_created and date_first_booking feature**, because a user can search on the website before creating an account and booking its first experience.
d. **date_first_book:** its date **(YYYY-MM-DD)** when the user did its first booking.
e. **gender:** It's simply the gender of the user. And we can have 4 values for this feature **"-unkown-", "male", "female", "other".**
f. **age:** Simply age of the user. But we do have many outliers in this because currently, **max-age in this feature is 2014** and **minimum age 1**, which clearly should not be the case.
g. **signup_method:** It means which method a user used to signup on Airbnb, whether he used **Facebook, Google, or simple email signup.**
h. **signup_flow:** This feature represents the number of pages a user came from to the signup page.
i. **language:** Simply the preference of the International Language of the user.
j. **affiliate_channel:** method of advertisement used with the first marketing provider with user interacted.
k. **affiliate_provider:** It's just the name of the affiliate provider or in simple terms by whom the user got to know or did sign up on Airbnb. L**ike, Google, Facebook, direct, etc.**

l. **first_affiliate_tracked: it shows the first marketing provider which brought users to sign up.**
m. **signup_app:** this feature tells the software platform on which the user has signed up, **like Web, ios, Android, etc.**
n. **first_device_type:** It shows the Physical Platform on which the user was first active, **like Mac Desktop, Windows Desktop, iPhone, etc.**
o. **first_browser:** The browser name on which users first visited Airbnb. **Like Chrome, Safari, Mozilla, etc.**
p. **country_destination:** This is the **country name** where the user has done their booking.

2. **test_users.csv:** This CSV contains the data for testing and cross-validation and has the **same features as our training data** except for our target variable which is "country_destination". For a feature explanation please look in the train_users.csv section and we have **62096 total data points** for testing.

   Also, one important thing to remember is, **we are not given any value for the "first_date_booking" feature in the test data**, so while building and testing our model we need to keep that in mind, and probably dropping that feature. Also we can see some basic information in below screenshots, like what are the **fields with NaN values, total_number of data points,** type of fields etc.



```
1 test_users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62096 entries, 0 to 62095
Data columns (total 15 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       62096 non-null  object
 1   date_account_created     62096 non-null  object
 2   timestamp_first_active   62096 non-null  int64
 3   date_first_booking       0 non-null      float64
 4   gender                   62096 non-null  object
 5   age                      33220 non-null  float64
 6   signup_method            62096 non-null  object
 7   signup_flow              62096 non-null  int64
 8   language                 62096 non-null  object
 9   affiliate_channel        62096 non-null  object
 10  affiliate_provider       62096 non-null  object
 11  first_affiliate_tracked  62076 non-null  object
 12  signup_app               62096 non-null  object
 13  first_device_type        62096 non-null  object
 14  first_browser            62096 non-null  object
dtypes: float64(2), int64(2), object(11)
memory usage: 7.1+ MB
```

3. **sessions.csv:** sessions.csv file contains the web sessions data. which basically explains what **users did (like what user searched, clicked, typed, etc) on the website** throughout the time spent on the website. Sessions CSV file contains **10567737** records which means we may have more than one session record for users and **somehow we have to make 1 row for each user id** without giving up valuable information to join with training or testing data. Here we have a **user_id column which corresponds to the id feature in Train and Test data.** And it only contains data of users starting 2014.

   This file contains **6 features** which are explained below. Also we can see some basic information in below screenshots, like what are the **fields with NaN values, total_number of data points,** type of fields etc. Here we can see we have Null values in user_id column as well, so we need to keep that in mind to drop these values, because there is nothing we can do about if user_id is not given.

```
1 sessions.isnull().sum()
```

```
user_id           34496
action            79626
action_type     1126204
action_detail   1126204
device_type           0
secs_elapsed     136031
dtype: int64
```

```
1 sessions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10567737 entries, 0 to 10567736
Data columns (total 6 columns):
 #   Column         Dtype
---  ------         -----
 0   user_id        object
 1   action         object
 2   action_type    object
 3   action_detail  object
 4   device_type    object
 5   secs_elapsed   float64
dtypes: float64(1), object(5)
memory usage: 483.8+ MB
```

   a. **user_id:** same as “id” feature in training and testing data, and this is given in this sessions.csv file **just to join the with train and test data** and use for Analysis and Modeling.
   b. **action:** this column or feature refers to action taken at the backend by the website.
   c. **action_type**: it’s just a type of action that was triggered at the backend by the website.
   d. **action_detail**: This feature shows the detail of action which is basically, happens when an action triggers.

e. **device_type**: It shows the Physical Platform on which the user was first active, **like Mac Desktop, Windows Desktop, iPhone, etc.**
f. **secs_elapsed**: total time in seconds elapsed since the last recorded action.

4. **countries.csv:** this CSV file shows the summary statistics of destination countries in the training dataset and their locations. This file is having 10 records and 7 columns with No NaN values which are as follows:
    a. **country_destination**: This column shows the **country code.**
    b. **lat_destination**: It shows the **Latitude value** of the country.
    c. **lng_destination**: It shows the **Longitude value** of the country.
    d. **distance_km**: This column shows the distance of all other places (Longitude and Latitude) from **Copan, OK, USA** in Kilometers.
    e. **destination_km2**: It shows the **Area of the country.**
    f. **destination_language**: It's simply the language of the country.
    g. **language_levenshtein_distance:** It's the Levenshtein distance **between eng and other mentioned languages**.

5. **age_gender_bkts.csv:** This CSV file shows the summary statistics of users' age group, gender, country of destination, In this file, **it would make more sense if we try to analyze all these features as a whole because individual features would not make any sense here.** We have 420 data points and 5 features with no NaN values, and the description of these features has been given below.
    a. **age_bucket**: This feature shows the age window of people.
    b. **country_destination**: This column shows the **country code.**
    c. gender: Simply shows gender.
    d. **population_in_thousands**: Shows the population of the people in thousands of mentioned age groups in the mentioned country.
    e. **year**: represents the country in which the above point d is applicable.

6. **Sample_submission.csv:** For every user in the dataset, submission files should contain two columns: id and country. The destination country predictions must be ordered such that the most probable destination country goes first.

Performance Metric

As our business problem is asking to predict 5 countries for each user and rank them according to their relevance or correctness, we have multiple Metrics we can use to validate that rankings, **like MRR**

(Mean Reciprocal Rank), MAP (Mean Average Precision), NDCG (Normalized Discounted Cumulative Gain).

First, let's see a little about the metric given by Kaggle to evaluate our model, we have given **NDCG** (Normalized Discounted Cumulative Gain) which is basically computed using **DCG (Discounted Cumulative Gain) and IDCG (Ideal Discounted Cumulative Gain)**. DCG is used in **Information retrieval to measure the quality of ranking**, and it is mostly used in web search engines to check the effectiveness of the algorithm.

So, as we know DCG is used to measure the **quality of ranking,** then why we are using NDCG here, the answer is simple, as **we are asked to predict 5 countries for each user and rank them** according to their relevance or correctness, **so to check that correctness or reliability of rankings we need a metric** and that's why we're using NDCG in this Business Problem.

NDCG will give value between 0 and 1 where 1 being perfect solution, so let's how do we calculate NDCG score

$$NDCG_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)}$$

Where k is number of predictions, rel_i is either **0 for wrong prediction or 1 for correct prediction,** for **example,** we have correct prediction US and we have predict it be most probable (means at top of 5 predictions) then $NDCG_k = \sum_{i=1}^{k} \frac{2^1-1}{log_2(1+1)} = 1.0$, now suppose we predict US at position 3 [**FR, AU, US**], in that case our $NDCG_k = \sum_{i=1}^{k} \frac{2^1-1}{log_2(3+1)} = 0.5$. **Now another case is of wrong Prediction**, means we are failed to predict US in top 5 predictions (**let's we predict US at position 6) then** rel_i **would be 0**, and this whole term would be $NDCG_k = \sum_{i=1}^{k} \frac{2^0-1}{log_2(6+1)} = 0$.

NDCG score would be **calculated by the positions of correct predictions**, meaning it **does not matter how much probability you got**, the **correct ranking matters.** NDCG will give a value between 0 and 1, here 1 being perfect rankings.

Research-Papers/Solutions/Architectures/Kernels

1. Predicting Airbnb user's desired travel destinations

There are lot of things and insights I learned from the above article, this paper is written beautifully and has given thoughts on many points like, **Problem with datasets, possible improvements, and suggestions, etc.** few of the takeaways **that can help me to start or build my project** in a better way from the paper are as follows:-

a. We will be using **Supervised Machine learning** because we have a dataset with answers and a dataset with no answers.
b. **Algorithms should be fast** because we have a large amount of data to train.
c. As we want our algorithm to be very fast and considering we have a multiclass classification problem to deal with, **XGboost would be an appropriate choice as it's known for its superiority for speed.**
d. one of the main thing I understood from this article is, why XGboost is an appropriate choice, it is because the size of datasets is increasing rapidly in today's world, and we all don't have the leverage of using a large amount of memory to process the data, so we had to come up with the ideas that are memory efficient and XGBoost is one of them.
   XGboost uses many techniques to tackle memory issues and most noticeable is **Cache Aware Access and out-of-block Computation**, cache-aware access basically when we **allocate internal buffers in each thread,** where the **gradient statistics can be stored.**
   Out-of-block computation means when **we store data parts onto disk instead of keeping everything in memory.** Also, XGBoost automatically uses all available cores.
e. **date_first_booking feature from the training dataset would be of no use** because we won't be having this column in our testing data so dropping this field is best.
f. One of the most important things I learned from this article is the ways to deal with sessions.csv files as we have only around **35 percent data available of training data.** One thing we can do is we don't consider the sessions.csv file at all and train our model only with train_users.csv data and lose all the sessions data.
   Another approach is we continue with sessions and users data and drop all the user's data which are not in session data, and here we are losing a lot of user data.
   The most favorable approach is we use outer join on sessions and user data and put 0 where we don't have session data which will indicate that those users used no devices and don't have session data. And here we would be having a full dataset to train our model. And yet it doesn't give a guarantee to score high, we will talk about this later.
g. As we're using **NDCG for evaluation,** and we're doing 5 predictions for each user, NDCG will give a value between 0 and 1, and it will only care about the correct ranking given to countries, **quantity of probability of output doesn't matter as long as rankings are correct**.
h. The **Sessions data is more important** in predicting the destination country as in the first model, only data present in sessions file was taken from train users and when feature importance was plotted and from top 15 features all features were of actions except 3. And this shows how important sessions data is.

i. And when we Joined all the data from the Sessions and Train Users file, there was a **minor difference in the accuracy, we got less accuracy,** and even then most of the actions fields were in important features which shows that **adding extra train users without sessions data doesn't add much value.**

j. An improvement could have been like instead of feeding all the features into the model, **we can find important features and drop unnecessary features** and then feed into the model, it could speed the training process

2. Applying LightGBM and XGBoost to Predict Airbnb User Booking Destinations:

This article has contributed to my understanding of the project and data I am going to deal with, now all the knowledge and experience I learned from this article will surely help to start the project doing myself in a better and smarter way.

One thing that I have noticed in all of the articles and papers related to Airbnb country prediction Kaggle competition that they **all are using ensemble techniques** to solve the problem, and that makes sense as the **ensemble model is indeed powerful**, and this article is no exception, and in this article **LightGBM and XGboost** are used to tackle with the problem, few of the takeaways **that can help me to start or build my project** in a better way from the paper are as follows:-

a. We can Join and Training and Testing data to get a collaborative understanding of data.

b. By exploring the country_destination, it is found out that a large number of people either **decided to go to the US or decided not to go at all.**

c. Also, we can check whether Gender has something to do with, above point b, and we found out that distribution is almost the same. **(It means weather Gender has something to do with choosing country US or not go at all)**

d. Another point I understood from this paper is that we can check whether age distribution has something to do with choosing to travel, and most of the **users were aged between 25 - 40.**

e. In this paper the user has **used XGboost and LightGBM**, both techniques work very well as they both use **ensemble techniques.**

f. One interesting thing I found out with the article is that both XGBoost and LightGBM **work very well with default parameter settings**, but of course, if we tune parameters it will surely increase the performance.

g. There is a **Benchmark Classifier** set by this paper owner and he used Random Forest for the benchmark, but I believe **we don't need it** because we can already figure out the **benchmark score with Kaggle Leaderboard.**

h. Of Course, we need to create **our own function to Calculate the NDCG score** as shown in this paper as well despite the fact **NDCG API is also available in sklearn library.**
i. According to the Paper, **LightGBM has given an NDCG of 0.825** with the default setting, and **XGBoost has given 0.80 NDCG scores** with default settings, but of course, this scenario might change as we proceed with some parameter tuning.
j. In this whole paper, I found a maximum NDCG score of 0.83 more or less, but in other papers and articles, I have seen this hike up to 0.87 or 0.88.
k. Another fact or more of ideas specific to this project I learned from this paper is, as we have seen, country_destination distribution is uneven among the target countries, so we can use **oversampling these countries to make the distribution even** and deal with imbalanced data problems.
l. We need to be very careful with the data while doing EDA and preprocessing as we have **many irregularities in data.**
m. Improvement mentioned in the paper is **more of a computational fancy**, but I believe that should not be the case, **we should look at improvement more of techniques based**, features engineering, etc.

3. Medium Blog - How does Airbnb know where you are going to Book your First Travel Destination :

The above article is written beautifully, it shows almost every aspect of this Airbnb business problem and explains various ideas which can be used to solve the problem, also as this article is **more EDA heavy,** I will **not be writing EDA Points here**, I will explore data myself, few of the takeaways **that can help me to start or build my project** in a better way from the paper are as follows:-

a. The owner of the article has **made a web app to represent his work**, which will surely help me to design and **improve my own web app to showcase my work**, also this article will **help me write my own blog,** and of course, these are used apart from actual ML help.
b. Only **around 35 percent of training data is present in session data** which leads to confusion about which data we should drop or use, well first we will use whole training data only and later data with session data points only and see the performance.
c. Our data is highly imbalanced and so needs to be dealt with via any technique like **upsampling or downsampling etc**.

d. The owner of the article has built a **model with different ML techniques** and has shown the performance of each model with **Pretty Table**, and here also **XGBoost is taking the lead** and that explains why **everyone is using this algorithm.**
e. **This article is more of EDA heavy, and will surely help me to understand the different aspect of EDA I can perform on the data, also future work is given like more accurate hyperparameter tuning, different feature engineering techniques which will surely help my to focus my energy in the right direction, the author has also productionized his model on Heroku Platform.**

4. Medium Blog - Airbnb New User Bookings — Kaggle Competition :

This article written by **Puneet Chandra** is also informative and few of the takeaways **that can help me to start or build my project** in a better way from the paper are as follows:-

a. The data available in the Sessions.csv file is **only for 35 % of the Training data,** so now either we drop the 65 % of the Training data to train our model or **we don't use the Sessions.csv file at all.**
   So, I will be **training my model in both the cases**, with sessions data and without session data.
b. The article shows that when we use only training data we get a **relatively low score,** compared to when we use session data as well, and it also makes sense because, in our previous paper, I studied that a **lot of important features come from actions.**
c. Also, we need to keep in mind that **we have more than one record for one user in session data**, so we need to convert the data such as we have only a **single record per user.**
d. The author of the article showed that he **got 0.95 NDCG scores on training data**, but got **0.8817 NDCG for test data**, so it shows that model **was overfitted** and I need to be careful about overfitting my model.
e. We can also think about **using Bi-gram and Trigram,** which may increase the accuracy of the model. I will surely try this.
f. Other things in this Article are more like **technically key points of the EDA** and First Cut approach, those I will figure out myself, and most of the points are already covered in the above articles or papers so not writing them again.

5. Airbnb New User Bookings:

Above Kernel is **quite impressive,** it has performed basic steps to solve any ML problem like it has gathered information about the missing value, weather data is imbalanced or not,

performed **various EDA on features,** performed feature engineering, one-hot encoding, and at last but not least applied Xgboost algorithm to solve the problem. A few of the takeaways **that can help me to start or build my project** in a better way from the paper are as follows:-

a. In the year feature of **age_gender_bkts.csv** we only have one unique value 2015, so **dropping this feature would be a wise choice** as this doesn't add much value, and we have one less feature to think about.
b. One thing that slipped from my mind was that **use of Median while imputing the Null values** if we have outliers in our numerical features **instead of Mean is a wise choice**, but after reading this Kernel I just remembered it.
c. The author has used the **Chi-Square Test** to find out whether there is a relationship between Gender and country_destination, but I might be using s**imple EDA to answer this question.**
d. This owner of the kernel has literally **done an excellent job in EDA**, he/she has explored almost each and every aspect of the features and this **will surely help me to get the ideas** of what can be performed or even enhancing them.
e. Another Idea I liked is that, as we have **more than 50 Browsers category** used by users but the majority of users are using 5 categories, so we can **just create one other category and keep all less used browsers in that.**
f. Most of the points in the kernel are similar to other papers and articles, which I have already gone through and written in the above sections.

First Cut Approach

There would be countless ways to approach this business problem, but steps to solve this problem as per my understanding are as follows, and certainly, I will improve the same as its first cut approach.

1. So, The First and most important step is, to **understand the business problem clearly,** in our case we have explained our Business Problem in an earlier section.
2. Next, we need data to solve business problems, and in our case **data is provided by Kaggle** as this is a Kaggle competition we're trying to solve, and all the data files have been already explained in earlier sections.
3. Now, I will try to explore the datafiles given and try to find out **how I can relate and use other datafiles like Sessions, Countries, etc with my training data file.**
4. Now would perform **EDA and preprocessing** on the training data, will try to find out correlated features and **information about missing values**. We have 16 independent features in our training data and 6 features in session data, we will explore each feature.

a. We will try techniques to deal with missing values or NaN, **like average values, more frequent values, or model-based techniques,** or in some cases will remove records with Null values.
b. Also, we will find out which feature is important in predicting the target variable and will **remove unnecessary features.**
c. We will also try to **find out the outliers** in our data and will deal with them, as we already explained earlier we have outliers in the Age column.
d. Also, we will do **feature extraction** for some columns. Like the date column is given as a string in our training data, **we will convert that into a date object**, etc.

5. The next step would be preparing our data.
    a. We can do some **feature engineering** in this section like we can come up with features l**ike days, month, years from first_account_active and date_account_created features**.
    b. Also, I will **deal with the Imbalance factor** of our data as we have already seen training data is highly imbalanced, so I will try to **upsample our data and see how it works.**
    c. Dealing with categorical features, like **converting them using one-hot encoding,** or using scaling technique for numerical features (though we **don't have much of numerical features in our dataset**)
    d. Now we will **split our data into training and testing data**, using sklearn's train_test_split API.
6. Now, when we have our data ready to feed into Machine Learning Model, we need to select our model, initially, I am going to try various models, and eventually will choose the best working model, models I am going to try are : **Logistic Regression, SVM, Random Forest, XGBoost, LightGBM.**
7. Finding the best Parameters are the most important and crucial factor to increase the model accuracy, so now we will find the best parameters for our model, **using GridSearchCV or RandomSearchCV**, I am gonna experiment with both, and will compare the time taken by both algorithms.
8. Now when I will be **having the best hyperparameters** for my ML Algorithms, I will be creating the model and will **check the NDCG score** and in order to check that I will be r**epeating all preprocessing and data preparation steps for my testing data as well.** And later will repeat the steps for improvements.