**Objective 1 and 2.**

**Issues:**

* For the assignment of languages ​​to the data of players, it was necessary to accept that player data can be repeated since in countries where it is common to use two languages, we do not have a parameter that helps us to weigh or choose one language. This also affects when we make clusters.
* For the analysis of countries there is a significant difference in the number of data between countries, with some with 1 or 10 records and others with more than one thousand, which makes it difficult to compare average between countries. This also affects when building clousters .

**Improvement opportunities**

* One solution of the language issue is to add a column with the mother tongue of the player.
* To achieve an improvement in the analysis between countries, it Will be necessary to have a data that guarantees a minimum of players per country, regardless of how low their performance is.
* The database of tweets was collected in streeming and for one hour only around 700 could be obtained and some of these are retweets, but we can assume that the user who retweeted shares the opinnon that is expressed there.

**Objective 3**

**Issues:**

* The tweet’s texts are short so the classifier is more susceptible to failures.
* Twitter locations are an open parameter so there may be different formats or description for one place, false locations, or coordinates.
* Googletrans is an API that fails when we make many requests, although a fake agent improves execution, this problem could not be avoided, so fewer comments were classified from those collected.
* The classifier used is trained with words from a context of movies and not video games so it may not be as effective, although the idea pursued in this test is to show knowledge.

**Improvement opportunities**

* Collecting the tweets daily and classifying them manually to build a classifier for these cases.
* It is possible to make a table or parameters that help us take the location and return a standard value.
* google-cloud-translate is a payment api with which you can solve the limit of requests.
* To build a more effective classifier, other categories such as neutral opinions or sarcasm could be included.