**Stage one of the Fractal AI Hackathon**

**Identify defective waterpoints in developing countries**

Water is one of the most basic human needs. Of late, we have been witnessing acute water shortages across countries and it is starting to become a critical problem in modern society. Governments spend millions in their attempt to ensure water availability across cities, towns and villages. Of all the water sources available, one source which stands out in ease of access as well as quality is ground water. In an urban setting, ground water is generally pulled via powerful pumps and stored in tanks. However, in rural areas wells and handpumps are the most accessible options. Governments install hand pumps in villages to provide for the villagers. However, often there are no regular maintenance checks of the pumps; partly due to high operational cost involved in travel and technicians, the maintenance of the hand pumps is generally ignored by the authorities.

In this data challenge, we ask:

Is there a way we can use predictive modelling techniques to predict whether a given handpump(read waterpoint) is defective? This would greatly help plan visits of technicians to places where problem exists instead of planning to go everywhere. There would be two direct benefits of this:

1. Huge cost savings
2. Early problem detection and resolution
3. Most importantly, people won’t have to wait indefinitely for help!

You have been provided with data collected across various villages in rural Africa. A lot of information has been captured related to the location, usage and attributes of the installed waterpoints. Given the features of an installed waterpoint, your goal is to classify a given water point as **defective** or **not.** Note that waterpoint includes hand pump, stand pipe, cattle trough etc.

**Data dictionary - the features in this data set are:**

* amount\_tsh - Total static head (amount water available to waterpoint)
* date\_recorded - The date the row was entered
* funder - Who funded the well
* gps\_height - Altitude of the well
* installer - Organization that installed the well
* longitude - GPS coordinate
* latitude - GPS coordinate
* wpt\_name - Name of the waterpoint if there is one
* num\_private - No information available
* basin - Geographic water basin
* subvillage - Geographic location
* region - Geographic location
* region\_code - Geographic location (coded)
* district\_code - Geographic location (coded)
* lga - Geographic location
* ward - Geographic location
* population - Population around the well
* public\_meeting - True/False
* recorded\_by - Group entering this row of data
* scheme\_management - Who operates the waterpoint
* scheme\_name - Who operates the waterpoint
* permit - If the waterpoint is permitted
* construction\_year - Year the waterpoint was constructed
* extraction\_type - The kind of extraction the waterpoint uses
* extraction\_type\_group - The kind of extraction the waterpoint uses
* extraction\_type\_class - The kind of extraction the waterpoint uses
* management - How the waterpoint is managed
* management\_group - How the waterpoint is managed
* payment - What the water costs
* payment\_type - What the water costs
* water\_quality - The quality of the water
* quality\_group - The quality of the water
* quantity - The quantity of water
* quantity\_group - The quantity of water
* source - The source of the water
* source\_type - The source of the water
* source\_class - The source of the water
* waterpoint\_type - The kind of waterpoint (handpump, communal standpipe, cattle trough etc.)
* waterpoint\_type\_group - The kind of waterpoint

**Labels (column name: Defective):**

* **yes** – the waterpoint is nonfunctional – this is the positive class in this problem
* **no** – the waterpoint is functional and there are no repairs needed

If a working waterpoint is wrongly classified as faulty (false positive), the cost involved might include a technician visit and inspection; however, if a non-functional waterpoint is classified as functional (false negative), then there is no action taken, and there is consequently a significant social cost. Hence, there is a much higher penalty for false positives.

Your model will be scored as follows:

**Weighted error rate = 0.9 \* FNR + 0.1 \* FPR,**

where FNR = false negative rate and FPR = false positive rate

**What to submit:**

1. Output: You should create a csv in the format given in the example below:

|  |  |
| --- | --- |
| id | defective |
| 101 | yes |
| 102 | no |
| 103 | no |
| 104 | yes |
| 105 | no |

Please name the csv in the format : **TeamMember1EmpID\_TeamMember2EmpID.csv**

Ex : If the Employee ID of Team Member 1 is F001 and 2 is F002 then the file is to be named as **F001\_F002.csv**. In case the team has only a single member name as **F001.csv.** Download the file to your local machine

1. **Code:** please submit the code used to generate your predictions. Accepted formats are .ipynb, .py and .R. Please read the triathlon rules for more information. Please name the code file in the same format as mentioned above. Ex. **F001\_F002.ipynb (or .py or .R).** Download the file to your local machine.
2. Put both files into a folder and create the zip in the same format as shown above. Ex. : **F001\_F002.zip**
3. Upload the csv on Hive at the submission bucket provided here :

<http://hive.fractalanalytics.com/course/view.php?id=433>