



Gotta Predict'em All!

Pokémon Go, a location-based, augmented reality mobile game released by Niantic Inc. The players use the GPS to locate, capture, and battle fictional creatures in a virtual setting who appear on the screen as if they were in the same real-world location as the player.

Like many millennials, I grew up watching Pokémon. So, when this game was released, I was undoubtedly excited, running on the streets of Jaipur looking for the rarest Pokémon's (but always ended up with common Pokémon's)

Although the game was a huge success, there were many flaws that needed to be addressed which ultimately caused the downfall of it.

Dataset:

Dataset consists of roughly 293,000 pokemon sightings (historical appearances of Pokemon), having coordinates, time, weather, population density, distance to pokestops/ gyms etc. as features. The target is to train a machine learning algorithm so that it can predict where pokemon appear in future. So, can you predict'em all?)

Feature description

- *pokemonId* - the identifier of a pokemon, should be deleted to not affect predictions. (numeric; ranges between 1 and 151)

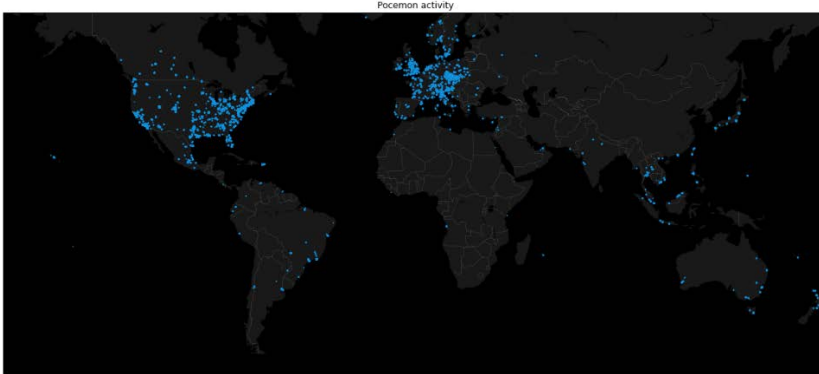
<https://pokedex.net/pokedex/national> for complete list of all Pokémons ids of gen 1.

- *latitude, longitude* - coordinates of a sighting (numeric)
- *appearedLocalTime* - exact time of a sighting in format yyyy-mm-dd'T'hh-mm-ss.ms'Z' (nominal)
- *cellId 90-5850m* - geographic position projected on a [S2 Cell](#), with cell sizes ranging from 90 to 5850m (numeric)
- *appearedTimeOfDay* - time of the day of a sighting (night, evening, afternoon, morning)
- *appearedHour/appearedMinute* - local hour/minute of a sighting (numeric)
- *appearedDayOfWeek* - week day of a sighting (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday)
- *appearedDay/appearedMonth/appearedYear* - day/month/year of a sighting (numeric)

- *terrainType* - terrain where pokemon appeared described with help of [GLCF Modis Land Cover](#) (numeric)
- *closeToWater* - did pokemon appear close (100m or less) to water (Boolean, same source as above)
- *city* - the city of a sighting (nominal)
- *continent* (not always parsed right) - the continent of a sighting (nominal)
- *weather* - weather type during a sighting (Foggy Clear, PartlyCloudy, MostlyCloudy, Overcast, Rain, BreezyandOvercast, LightRain, Drizzle, BreezyandPartlyCloudy, HeavyRain, BreezyandMostlyCloudy, Breezy, Windy, WindyandFoggy, Humid, Dry, WindyandPartlyCloudy, DryandMostlyCloudy, DryandPartlyCloudy, DrizzleandBreezy, LightRainandBreezy, HumidandPartlyCloudy, HumidandOvercast, RainandWindy) // [Source](#) for all weather features
- *temperature* - temperature in celsius at the location of a sighting (numeric)
- *windSpeed* - speed of the wind in km/h at the location of a sighting (numeric)
- *windBearing* - wind direction (numeric)
- *pressure* - atmospheric pressure in bar at the location of a sighting (numeric)
- *weatherIcon* - a compact representation of the weather at the location of a sighting (fog, clear-night, partly-cloudy-night, partly-cloudy-day, cloudy, clear-day, rain, wind)
- *sunriseMinutesMidnight-sunsetMinutesBefore* - time of appearance relatively to sunrise/sunset [Source](#)
- *population density* - what is the population density per square km of a sighting (numeric, [Source](#))
- *urban-rural* - how urban is location where pokemon appeared (Boolean, built on Population density, <200 for rural, >=200 and <400 for midUrban, >=400 and <800 for subUrban, >800 for urban)
- *gymDistanceKm, pokestopDistanceKm* - how far is the nearest gym/pokestop in km from a sighting? (numeric, extracted from [this](#) dataset)
- *gymIn100m-pokestopIn5000m* - is there a gym/pokestop in 100/200/etc meters? (Boolean)
- *cooc 1-cooc 151* - co-occurrence with any other pokemon (pokemon ids range between 1 and 151) within 100m distance and within the last 24 hours (Boolean)
- *class* - says which pokemonId it is, to be predicted.

- ❖ *Plot Pokemon activity data points based on latitude and longitude globally on the map with the help of matplotlibs basemap Toolkit.*

Sample image output:



- ❖ *Which cities have the highest Pokémon activities count. Plot a bar graph using seaborn and matplotlib.*
- ❖ *Perform Classification of Pokémon's(id) by continents latitude and longitude.*
- ❖ *Add a column with label 'names' and create a dict of all names from the pokemondb url which corresponds to the ids.*
(<https://pokemondb.net/pokedex/national>)
Now using map function , associate the names with the provided ids.
- ❖ *Predict where a Pokémon of your choice will spawn.*

Please make a proper report on this project and submit soft copy to Forsk official email id.

It should atleast contain the following SECTIONS:

- Cover page
- Certificate on Forsk letter head
- Acknowledgement
- Abstract
- List of Figures
- Table of contents
- Introduction
- Background theory/Motivation
- Methodology:
 - Data Pre-processing /Data Munging
 - discussion on Regression methods used
 - discussion on Classification methods used
 - (knn/logisitic)
 - Data Visualization.
- Result Analysis
- Conclusion.
- Reference [OPTIONAL]
- ANNEXTURE [WILL CONTAIN THE CODE HERE, don't include lengthy codes in the main part of the report, use code snippets for specific sections]

(Also include the code.py file along with the report in zip folder)