# Disaster Management Analytics

## Ashish Kumar, 4/6/2019

1. **Historical accident data: Capture, Storage & Usage**

**Summary:**

**Data Source –** Historical accident data from Meteorology department, Survey/Census data for socio-economic variables, Geolocation data from GPS Sensors

**Data format** – Spatial Polygon DataFrames

**Databases** – postGIS, PostgreSQL; .gri, .gpkg, .sqlite, .shp, .kmz, .mdb, ArcSDE as flat files on AWS S3

**Use Cases/Benefits**

1. **Disaster Budget Planning** - Monte Carlo Simulation analysis of disaster occurrence probability and potential loss evaluation. Would help in planning disaster budget and avoid under/over budgeting
2. **Disaster Relief Fund Allocation** – Match the zip/household level socio-economic data with the disaster impact (image/video) data to allocate the relief funds based on need rather than doling the same relief material/amount to everyone
3. **Identify location and time accident hotspots** – Run time-series analysis and animations on accident data laid over map and/or timeline to identify which locations and time of year are more prone to what kind of disasters

**Technologies/Algorithms** – Python, ArcGIS, R Shiny, D3.js, ARIMA time-series analysis, Monte Carlo Simulation, Clustering, Data Animation, Geo-mapping

* Spatial polygon dataframes\* format would be used to attach accident's geolocation to it's data e.g. types of disasters, number of casualties, severity scale, current weather data, zip-level or household level socio-economic data etc.  Accident data can be collected from local metrological departments and encapsulated in the spatial dataframes format.
* Create a ETL pipeline to read and process the historical accident data containing the spatial coordinates and boundaries.
* This pipeline would use R (a statistical tool) packages like 'sp' and 'sf' to convert the survey data to data structures amenable for geospatial analytics viz. spatial polygons, spatial lines, spatial dataframes
* Spatial polygon dataframes can be stored as physical files as .gri, .gpkg, .sqlite, .shp, .kmz, .mdb, ArcSDE or in a database like postGIS
* This data can be stored as layered or Raster data format as well for storing time-series geospatial data e.g. weather data over time for sites where accidents have happened in the past
* The pipeline would be usable for all the data till the time all the survey data is in the same format. If format is different then a new pipeline needs to be created.
* A simple UI would be created to push the collected accident data through the pipeline
* QA would be performed on the loaded data

*\*just capturing latitude/longitude is not enough for good analysis. Spatial polygon dataframe contain the boundary coordinates of the area which are needed for powerful analysis*.

**2) Real time climate and weather monitoring data: Capture, Storage & Usage**

**Summary:**

**Data Source –** Real time and historical image/video/raster/IoT data capture from Drones, Radars, IoT Sensors and Satellites

**Data format** – Images (jpg,tiff,png), Videos (mp4, avi, live stream), Raster (stack of GIS images), Time Series IoT Data (TDMS)

**Databases/Storage** – AWS S3, Hadoop

**Use Cases/Benefits**

1. **Video Analytics based disaster alarm** – Train a model to learn the image patters which might indicate a disaster in near future using Video Analytics techniques like YOLO, CNN etc. Run this model on live stream data to give in-time disaster alarm
2. **Monitor Eco-sensitive zones over time** – Monitor the water level, deforestation rate, degree of mining activity, unbridled urbanization, gentrification over time using image animation of pictures taken over time
3. **Sensor data-based disaster prone zone identification** – Run multivariate outlier detection based methods on IoT sensor data capturing variables like air and water pollutant levels, soil quality etc.

**Techniques / Algorithms** – You Only Look Once (YOLO), Convolutional Neural Networks (CNN), Mahalanobis Distance, ARIMAX, Survival Analysis

* Video capture using satellite and drones stored on AWS S3 or Hadoop
* GIS Images and Rasters collected by Satellites stored on AWS S3 or Hadoop
* IoT Sensors data stored in TDMS files or some time series database like Splunk

**3) Pre-existing alternate climate/GIS data: Capture, Storage & Usage**

**Summary:**

**Data Source –** Open and free data archives collected by government departments, private companies and research agencies and hosted in cloud. One example is [Earth on AWS](https://aws.amazon.com/earth/) repository. Few examples of stored data are – Sentinel 2, Landsat 2 (Satellite Imager data), Terrain Tiles (Contour Heights of bare Earth), SpaceNet (Labelled data optimized for machine learning based video analytics using computer vision techniques), Global Database of Events, Language and Tones (repository of Global Events), NEXRAD (real-time radar weather data)

**Database/Storage**- Stored on AWS or any other cloud storage

**Data format** – Images (jpg,tiff,png), Videos (mp4, avi, live stream), Raster (stack of GIS images), Time Series IoT Data (TDMS)

**Use Cases/Benefits**

1. **Research for Data Storage** – This research would help us understand what data is already available, what data is further needed to achieve what we want, in what format do we store collected data and where do we store them.
2. **Prototyping of machine/deep learning applications** – These free datasets can be used for making prototypes/proof of concepts of the applications to test their feasibility and accuracy. Also to get a sense of efforts and costs needed in scaling the applications.

**Phases/Components of the Project**

**Data Collection**

* Conduct surveys and primary research
* Install drones and IoT Sensors
* Contact govt agencies for satellite imagery feed
* Consult govt agencies to improve their satellite imagery feed
* Create Storage Infrastructure (Cloud Storage @fixed cost per month)
* Compile a list of open source alternative data sources

**Data Engineering**

* Define formats to store the different data sources
* Decide the right database to store each kind of data source
* Create a frontend UI to send the collected data to the database
* Create a backend ETL pipeline to store the collected data to the database
* Run Quality Assurance on the collected data

**Data Science/Machine Learning/Deep Learning**

* Identify and prioritise use case(s)
* Data labeling if the application involves supervised machine/deep learning applications
* Conduct Exploratory Analysis based on the data pertaining to the application
* Feature Engineering to identify important features/variables in the data
* Training the appropriate model for prediction
* Fine tune and optimize the model parameters for them to generalize on any data
* Store the model as a serialized physical file to be reused later

**Productionizing the application**

Make an API/web application/product/report for the user to consume our model or analysis