**Report – Project A**

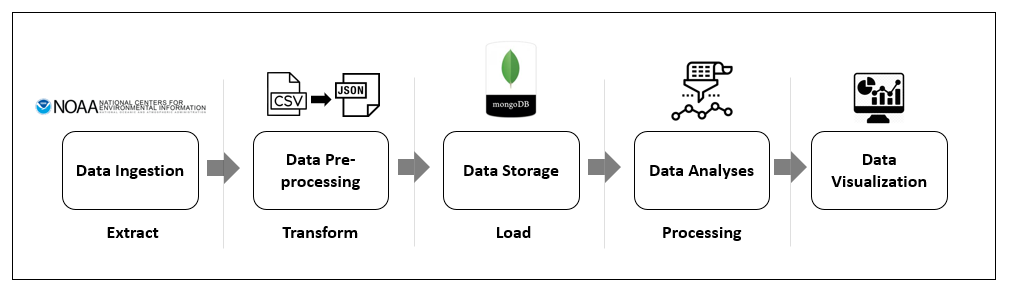
**Working with National Weather Service Data**

**Introduction**

The National Weather Service Data contains data from January 1950 to July 2019. The data is available as CSV files. The analysis done here covers the period 2012 – 2017. In terms of Event Types, this analysis focuses on Tornados. The process followed for this analysis manually simulates a standard pipeline. The pipeline was executed in a Jetstream virtual machine. An Instance of MongoDB 4.2 database was installed. Also, Python (version 2.7) along with Pandas and Pymongo libraries were installed in the VM for doing this analysis. The visualizations were created using Plotly on Jupyter Notebook as well as Tableau (both on a local machine)

**Pipeline**

A standard pipeline methodology as depicted below was followed for doing this analysis



The stepwise process followed is tabulated as below:

|  |  |  |
| --- | --- | --- |
| **Pipeline stage** | **Tasks done** | **Tool used** |
| Data ingestion | The data dump was downloaded from the NOAA website | Python utility file |
| Data extraction | Extract the CSV file from the original GZ format | Python utility file |
| Transform | Convert the SCV file to JSON for loading into MongoDB. To keep the data size small, a subset was queried by selecting specific columns | Python utility file, Query |
| Load | The JSON file is now loaded on to the MongoDB database | Python utility file |
| Data analysis / processing | The selected dataset was queried to filter out “Tornado” events and exported out as a CSV file | Query script |
| Visualization | The exported data was used to generate visuals | Plotly & Tableau |

**Visualization 1**

A close up of a map

Description automatically generated

The above graphic captures the state-wise direct death toll from tornadoes during the study period. Oklahoma with 45 deaths has the highest casualty followed by Mississippi (30), Texas (25), Kentucky (22).

**Visualization 2**: A more detailed breakup of the relative death toll state-wise is captured in the tree map below: The detailed statistics in the Tableau visual below is available as a “hover over”.

A screenshot of a cell phone

Description automatically generated

**Visual 3**

A close up of a map

Description automatically generated

**Visual 4**

A close up of a map

Description automatically generated

The above graphic captures the property damage on account of tornadoes. Texas, Illinois and Oklahoma contribute the most. This correlates with the direct death toll of these states.

**Visual 5**

A close up of a map

Description automatically generated

The state-wide count of tornadoes is shown above. Quite predictably, Texas, Oklahoma, Kentucky and Mississippi figure in the high-count states.

**Visual 6:**

A close up of a map

Description automatically generated

This graphic speaks to the count of significant, defined as EF Scale >=2, state-wise. The same set of states are at play here again.

**Challenges:**

The implementation was relatively smooth except for a few minor glitches:

1. The Python utility file had some errors before it was replaced.
2. I had unknowingly deleted the VM instance while changing the configuration from ‘tiny’ to ‘small’ to accommodate the larger size of Hadoop and Spark files. The instance was however recreated.
3. Had some issues initially toggling between MongoDB and the Linux consoles.
4. Since I had not figured out a way to copy-paste on to the Atmosphere command prompt, had some challenges typing in long commands without errors. However, this was resolved by installing Putty through WinSCP.

**Suggestions**

I found the process of using Python scripts as a standalone utility to extract and download somewhat prone to problems. Perhaps a more robust alternative could be found, although I am not able to comment on what it could be.

**Resources used**

1. Dataset : <https://www.ncdc.noaa.gov/stormevents/details.jsp>
2. Python utilities script
3. VM image on Jetstream pre-installed with MongoDB, Python, Pandas and Pymongo libraries
4. Jupyter Notebook
5. Tableau