

# Seismic Event BI Dashboard

## 1. Introduction

Our lives are impacted by natural disasters. In New Zealand (NZ), one of the top natural disasters that has the most impact on us and our properties is earthquakes. Imagine working at an insurance company that covers natural disaster claims. This project aims to build a Business Intelligence (BI) dashboard to assist in analysing the impact of major earthquake events on properties in NZ, visualise whether a claim address is impacted, and whether a claim address has been impacted by other events previously, which may have one or more prior claims for fraud analysis.

## 2. User manual

The entire project's source code has been integrated into one Python file, "seismicEventVisual.py". It consists of four classes and runs into two different functional programs controlled by the run arguments passed by the user. Before starting to run the programs, the current working directory needs to be set as "src" in the "workspace" folder, where the source code is located.

### 2.1 Event data file creation

By passing a run argument "-f" in debug mode, one of the programs will be triggered to source the seismic events data, with one event per row, that exceeds the magnitude threshold passed as a parameter to the "Source\_History\_Event" class. The events are within the date range input by the user, who is prompted to input start and end dates in a restricted format at the beginning of the program run. It also combines the affected polygon geometry, derived from calculations based on the associated station data of a given event, and then outputs a CSV file with 25 columns. The file comprises the following data attributes:

- Column A, public ID, the unique identifier of each event.
- Column C, event origin time, in NZ's local time zone.
- Columns S and T, coordinates of the epicentre.

- Column V, Modified Mercalli intensity (MMI) scale, a better indicator of an earthquake's effects on people and their environment, taking into account both depth and magnitude (*GeoNet Earthquake Intensity*, n.d.).
- Column Y, affected polygon, exceeding the boundary magnitude threshold given as a parameter.

## 2.2 Event map visualisation

Once it has completed successfully and the output CSV has been stored in the "data" folder, which is in the same "workspace" folder as the source code, it will prompt a message that includes the file name. The file name (includes extension) can then be copied and passed to the class "Event\_Contour\_Map" as a string parameter to its function "read\_gdf\_from\_wkt", enabling the second program to run in debug mode with the run argument "-p" passed.

The second program opens a GUI window created by the PyQt5 Python library. The GUI window consists of two sections. The left section comprises a 720 by 800 window that plots an interactive Folium map, visualising the chosen seismic event with the following layers:

- NZ territory area information.
- The epicentre location of the chosen event as a drop pin.
- The associated stations that recorded the MMI scale readings, shaped as towers.
- The intensity contour lines in gradient colour, showing the impact area and corresponding severity defined by MMI.
- The affected polygons, circled in green, that exceed the given boundary magnitude threshold.

The searchable dropdown box below allows user to choose an event by its unique identifier, public ID, and the push button, labelled as "Show Event Density Map", plots the event on the map. A brief metadata of the event is also provided at the bottom of the left section.

In the section on the right, an equal-sized Folium map visualises the chosen address location, shaped as an orange house, and its associated historical seismic events, included in the CSV file sourced previously, as drop pins. The definition of a seismic event associated with the address is whether the coordinates of the address are within the affected polygon of the event. The

searchable address-choosing dropdown box and "Show Address History Events" push button are right below the map.

The Folium maps can be zoomed in and out, controlled by mouse or "+" and "-" buttons on the top-left of the maps. The maps can be moved around, and layer boxes are on the top-right of each map to control the visibility of each map layer. In the event density map, on top of the layer box, a contour line colour-map legend is also provided to reference the gradient colour with its corresponding MMI scale. On the top-left of the main window, a "File" button in the menu bar has been created for the user to close the dashboard by clicking or a shortcut.

### 2.3 Business logic

The dashboard is useful when a natural disaster insurance company received a claim, the claim assessment team can then show the address of the property in the section on the right, to visualise whether the property is impact by the claimed event, or whether there is any event occurred close to the time of the loss. If the associated event that potentially caused the loss has been identified, it can then be shown in the left section on the map to visualise the detail of the event to assess how severity of the impact. In the scenario if there is no associated event is shown, one of the following options could be useful.

- If there is a claimed associated event, it then can be visualised in the left section to address the likelihood of an impact.
- Lower the magnitude thresholds of both event and boundary (for the affect polygon) in the event sourcing program to re-source the event data. Then come back to the dashboard to visualise whether there is an associate event and analyse the relationship between the event and the loss.

The section on the right is also useful for identifying whether there are any associated historical events that might have had an impact on the property, to address whether a prior claim occurred for further fraud analysis.

## 3. Development

At the initial phase of the development, the following data were downloaded to start with:

- NZ address CSV file from Land Information New Zealand (LINZ) (*LINZ Data Service*, n.d.).
- Historical seismic event data from GeoNet Web Feature Service (*GeoNet Web Feature Service*, n.d.).
- Nested event data from GeoNet API, such as station data and MMI scale for a specific event (*GeoNet API*, n.d.).

After the data were sourced successfully, an idea emerged to plot event epicentre locations and intensity contours to show the impact of an event after seeing the shaking layer map on GeoNet (2023p310616, n.d.). The Inverse Distance Weighting (IDW) method was used in the calculation of the value  $z$  based on the event and station data, with coordinates as the spatial values and MMI scale as the original  $z$  values (*Inverse Distance Weighting (IDW)*, 2017). With scaled-up spatial values, the  $z$  values were estimated, and the density contour was plotted with Matplotlib, originally shown in Figure 1.

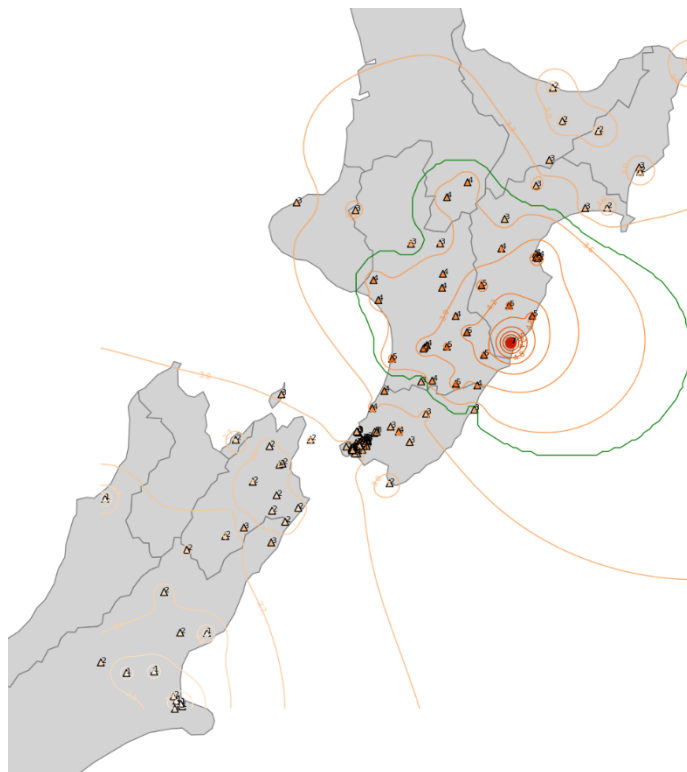


Figure 1: Public ID 2023p310616 event density contours plot in Matplotlib

In an attempt to match the contours with the map shown on GeoNet, with Figure 2 as an example, many experiments were conducted with different IDW parameters, namely "power" and "epsilon", and different weightings of the epicentre MMI.

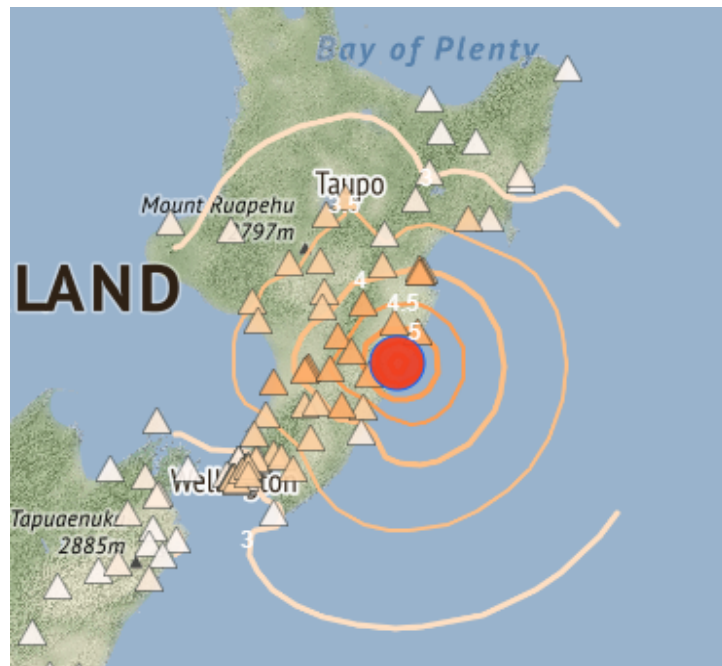


Figure 2: Public ID 2023p310616 event density contours on GeoNet (2023p310616., n.d.)

To enhance the visualisation with a more realistic appearance, flexibility to control the zoom and level of detail and comprehensive information provided with tooltips, the Folium map Python library met those expectations after conducting a search online. The result of the same event is shown in Figure 3.

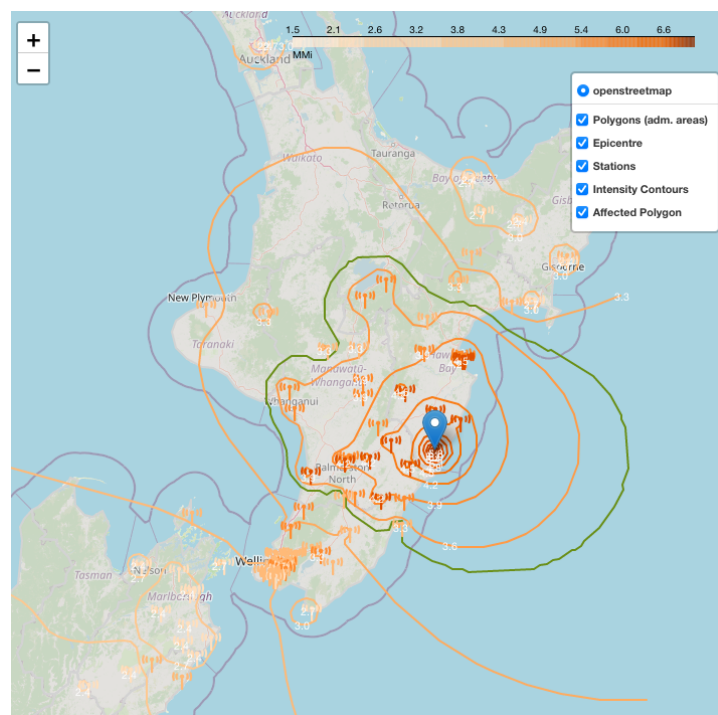


Figure 3: Public ID 2023p310616 event density contours and affected polygon plot in Folium.

The next question was how to determine whether a property had been affected by an event. The idea came up was to use a cut-off MMI threshold for the boundary of the affected area of an event. As shown in Figures 1 and 3, the affected polygons circled in green have a threshold of 3.5, which means that an event had an impact on a property if it is within the 3.5 MMI area. To display multiple historical events related to a single property, the affected polygons for each event must be calculated through a complex process every time a new address is selected. To avoid redundant calculations, the estimated affected polygons are pre-processed and stored in a CSV file, rather than storing them on an instance basis. A dedicated class, `Source_History_Events`, has been developed to create a dataframe utilising Pandas and GeoPandas Python libraries, and then save it as a CSV file stored in the "workspace/data" folder. A batch processing system has been developed to efficiently process long date ranges across multiple years, handling a high volume of events and providing enhanced debugging insights.

Once the CSV file is stored successfully, the Folium map to visualise property addresses and their associated historical seismic events is developed in a different class, "`NZ_Addresses`". The address data comes from the downloaded LINZ NZ addresses data, and the associated historical events data is filtered by scanning through the affected polygon geometry data in the CSV file to determine whether the property address is within any of the polygons.

By storing the event metadata and affected polygon geometry data in a CSV file, the density contour Folium map processing is more efficient, implemented in the class, "`Event_Contour_Map`". This eliminates the redundant sourcing of events' metadata and estimated affected polygons. This class returns a Folium map that plots a single event, its density contours, associated stations, and affected polygon in different map layers.

Finally, both Folium maps created in the "`NZ_Addresses`" and "`Event_Contour_Map`" classes are presented in a GUI window as a dashboard created using the PyQt5 Python library and integrated into the class, "`Ui_MainWindow`". The idea of using PyQt5 was researched online, and the results show that it works seamlessly with Folium maps. The two Folium maps are represented side by side to facilitate comparison.

#### 4. On track

During the development process, things went well as follows:

- GeoNet provided comprehensive information and examples that made sourcing seismic event-related data through its Web Feature Service and API simple. This included passing extra parameters such as time, magnitudes, and coordinates that limited the events to those that occurred not far from the shore. Dictionaries and dataframes were utilised to unpack the returned JSON-formatted data.
- The coordinate reference system (CRS) used in the basemap of the Folium map and the NZ addresses data downloaded from LINZ were in European Petroleum Survey Group (EPSG) 2193, while data sourced from GeoNet was in EPSG: 4167. The GeoPandas library enabled CRS conversion when converting the dataframe to a Geodataframe, making this conversion simple.
- The calculation of the z-value using the IDW algorithm was straightforward once the logic of the calculation was understood. Ensuring the shapes of NumPy arrays were correct was critical in every step of the calculation.
- Plotting a Folium map was simple, as it came with a realistic looking basemap with an interactive system and tooltip enabled. Users only needed to add extra layers on top of the basemap, and multiple layers could be grouped to form a feature group. There were plenty of online resources available for learning.
- The usage of the PyQt5 library to create a GUI window was intuitive, with fruitful online resources and videos available to learn from.
- Integrating both source event data and visualisation programs into one Python file went straightforward. By doing this, the overlapped functions between “Event\_Contour\_Map” and “Source\_History\_Events” could be removed from one of the classes and reused from the other class. The run argument was useful in this case, allowing the program to run into different modes by simply looking up the keyword included in the system arguments.

## 5. Challenges

As mentioned previously, attempting to match the density contours of seismic events to reflect the impact of the events with the shaking layer map on GeoNet was a challenge. This involved many experiments of parameter tuning for the IDW algorithm to make them as close as possible. However, the station MMI scale data precision was different. The station MMI scale readings sourced from the API were integers, while they were floats with one decimal place

shown on the shaking layer maps on GeoNet. The data precision limited the accuracy of the intensity contours derived from IDW. Additionally, there were multiple affected polygons derived from a single event, as the polygon boundary MMI scale was 3.5, such as public ID 2023p310621 and 2024p063773. Those events only had one 3.5 MMI contour line on GeoNet's website. When a station with an MMI above the threshold was far away from other stations, it formed a separate polygon. It could not generalise the scenario well by connecting the contour lines to form one larger polygon.

Another challenge arose when forming the affected polygon based on the edge of the coordinates; it could not deal with incurves. When one station was tucked in between its neighbouring stations, it would be avoided by connecting the neighbouring stations together with a straight line. After online research, a concave hull algorithm was sourced from GitHub that dealt with concave-shaped polygons (Dwyer, K, 2014). The function "alpha\_shape" in the "Event\_Contour\_Map" class was referenced from the Jupyter Notebook online.

The size of the original NZ addresses CSV file downloaded from LINZ was 784 MB, consisting of over a million rows. Loading data from such a large file into memory severely impacted the speed of the visualisation. The workaround was to only include a subset of the file to speed up the program.

To filter historical seismic events correlated with a selected NZ address, it needs to be determined whether the address's coordinates are within the affected polygon from column Y in the sourced event CSV file or whether the affected polygon contains the geometry point of the address. Either the "contains" or "within" functions from the GeoPandas Python library could have limited efficiency on performance, especially when dealing with a large amount of data. For instance, filtering NZ addresses within an event's affected polygon by scanning through the full NZ addresses CSV file to check whether the geometry point is within or contained would be slow. Instead of looping through, the "sjoin" function from GeoPandas has greater efficiency and performs much faster with an inner join, setting the parameter "predicate" equal to "within". It uses spatial indexing, e.g. R-tree, which allows GeoPandas to quickly narrow down potential matches, leveraged vectorisation that processes multiple geometries in a single operation using optimised low-level functions, and parallel processing boosts the performance (Pacull, 2021).



Due to the incomplete and inaccurate station data in historical events, especially in the 2010s, the program will run into errors when the boundary MMI threshold is set lower, such as below 3. For instance, public ID 3134797 and 3072353 each have only one station record, and both stations' MMI readings are 4. This also made the intensity contours in the early years inaccurate.

## 6. Future improvements

The programs can be improved as follows:

- Complete the menu bar by adding more functions to the "File" menu, such as saving maps, and add other menus, such as "View" and "Tool" menus, to enrich the dashboard's functionality.
- Make the GUI window elastic when resizing the window.
- Add brief property metadata, including year built, land elevation, land Technical Categories (TC), exterior material, or core claim data, to the bottom right section of the dashboard to aid in claim assessment.
- Improve the dashboard's robustness by displaying maps with a prompt warning when no event or address is selected, rather than encountering errors.
- Make the source event data program and plot event density contour Folium map functions resilient to low boundary MMI thresholds, even when station data is unavailable. Instead of encountering errors, the programs should either ignore such events during data sourcing or prompt warnings to select an alternative boundary MMI threshold for plotting the event.

## 7. Conclusion

In conclusion, there is still a significant way to make this application more user-friendly and robust. Most of the time invested has focused on ensuring the intensity contours accurately reflect the true impact, aligning with the project's primary objective. Providing claim assessment agents with more useful tools will enhance the efficiency and accuracy of the claims process, mitigating the impact of natural hazards on individuals and enabling informed decision-making. Additionally, this will help prevent fraudulent activities and enhance the claim validation process.

## Works Cited

2023p310616. (n.d.). Wwww.geonet.org.nz. Retrieved May 23, 2024, from

<https://www.geonet.org.nz/earthquake/felt/2023p310616>

Dwyer, K. (2014). *concave hulls using shapely and scipy*. Gist.

<https://gist.github.com/dwyerk/10561690>

GeoNet API. (n.d.). Api.geonet.org.nz. Retrieved May 22, 2024, from

<https://api.geonet.org.nz/>

GeoNet Earthquake Intensity. (n.d.). Wwww.geonet.org.nz.

<https://www.geonet.org.nz/earthquake/intensity>

GeoNet Web Feature Service. (n.d.). Wfs.geonet.org.nz. Retrieved May 22, 2024, from

<https://wfs.geonet.org.nz/>

Inverse Distance Weighting (IDW). (2017, October 18). Wwww.geo.fu-Berlin.de.

[https://www.geo.fu-berlin.de/en/v/soga-r/Advances-statistics/Geostatistics/Inverse-Distance-Weighting-](https://www.geo.fu-berlin.de/en/v/soga-r/Advances-statistics/Geostatistics/Inverse-Distance-Weighting-IDW/index.html#:~:text=The%20inverse%20distance%20weighting%20(IDW)

[IDW/index.html#:~:text=The%20inverse%20distance%20weighting%20\(IDW](https://www.geo.fu-berlin.de/en/v/soga-r/Advances-statistics/Geostatistics/Inverse-Distance-Weighting-IDW/index.html#:~:text=The%20inverse%20distance%20weighting%20(IDW)

LINZ Data Service. (n.d.). Data.linz.govt.nz. Retrieved May 22, 2024, from

<https://data.linz.govt.nz/layer/105689-nz-addresses/>

Pacull, F. (2021, September 10). Spatial Join with GeoPandas (and GEOS). Architecture &

Performance. <https://aetperf.github.io/2021/09/10/Spatial-Join-with-GeoPandas-and-GEOS.html>