Machine Learning Engineer Nanodegree
Capstone

Kyle McMillan
January 27th, 2018

Earthquake Prediction

I. Definition

Project Overview

In September 2010 at 4:35am local time, there was a large magnitude of 7.1 earthquake in Christchurch, New Zealand. This caused widespread damage throughout the city but there were luckily no fatalities linked to it. On February 22nd at 12:51pm there was another large earthquake, this time measuring 6.3. This cause even more destruction to an already devastated city and cause the deaths of 181 people. While this is just one case of one city, there are many cities all over the world that have been hit by large earthquakes. Larger ones don't only affect one city and can leave problems for the whole country, or even other countries.

Every year, earthquakes case millions of dollars in damages to property, but more importantly, earthquakes can have high death tolls. For example in 2004 there was 298,101 deaths worldwide from earthquakes[1].

With the ability to be able to predict when an earthquake may appear can give people time to move. If there is a tsunami involved, such as in the Indian Ocean earthquake[2], an early warning can give people time to move to higher ground. Or give an incentive to build stronger structures in an area where a strong earthquake could be predicted to appear.

In general it is believed that earthquakes are impossible to predict. Therefore the aim of this project is to see if there is any truth in that belief. This project, will focus on a single country's earthquake information: New Zealand. The reason for using New Zealand is because it is a country located within the Pacific Ring of Fire[3] and has approximately 10,000 earthquakes occur every year. Using this information, the project will utilise a machine learning technique called random forest to look for relationships in the data and see if those relationships can lead to earthquake predictions.

Problem Statement

The project will use earthquake data (timing, depth, and location (longitude and latitude)) to predict if an earthquake is greater or smaller than a 5.0 magnitude. Therefore, this task is a binary classification problem that will use a supervised learner to investigate the solution.

Metrics

Since the output of this project will be a binary classification, and the data is also imbalanced, an f-score evaluation will be used. The project aims to have recall over precision but precision is still important. The reasoning for this is because a false-positive is better than a false-negative; it's better to prepare for earthquake that doesn't come over not preparing for an earthquake that does come. The beta value will be 1.25.

---

1 Statista, (n.d.). "Global death toll due to earthquakes" Retrieved from https://www.statista.com/statistics/263108/global-death-toll-due-to-earthquakes-since-2000/

2 *In wikipedia*. "2004 Indian Ocean earthquake and tsunami" Retrieved 2018/01/27, retrieved from https://en.wikipedia.org/wiki/2004_Indian_Ocean_earthquake_and_tsunami

3 *In wikipedia*. "Ring of Fire" Retrieved 2018/01/27, retrieved from https://en.wikipedia.org/wiki/Ring_of_Fire

II. Analysis

Data Exploration

The dataset is from GeoNet[4]. The downloads are free and this project will use the "All Quakes in CSV format" download. The raw dataset contains 587,333 (as of 13/01/2018) earthquakes with 21 features. A lot of these features contain information about how the magnitude or location of the earthquake was calculated. Since these are not relevant to the project these are planned to be removed from the dataset. Since the information, within the dataset is based upon the seismic sensors and measure vibrations, there are some data points that are listed, but are not due to earthquakes; landslides and nuclear testing, for example. Events not listed or assumed as earthquakes will also be removed.

Raw data (before cleanup)

- Total event points: 587,333
- Number of features: 21
- First recored event: 1460/01/01 00:00:00 (Data gathered from felt reports)
- Most recent recorded event:2018/01/13 11:32:01
- Data points with magnitude less than 5: 583,749
- Data points with magnitude greater than or equal to 5: 3,584
- Event rate: 0.0061 or 0.6%

With an event rate of 0.6% it means the data is heavily imbalanced and will require special techniques, such as over-sampling, to compensate for the low event rate.

Exploratory Visualization

A sample of the data before processing:

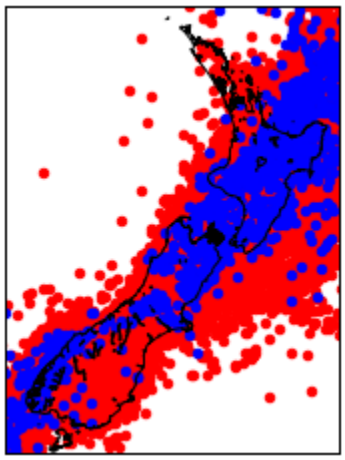
	publicid	eventtype	origintime	modificationtime	longitude	latitude	magnitude	depth	magnitudetype	depthtype	...
0	2018p033731	NaN	2018-01-13T11:32:01.243Z	2018-01-13T11:33:51.276Z	173.786363	-42.207544	2.173616	25.156250	M	NaN	...
1	2018p033726	NaN	2018-01-13T11:29:52.885Z	2018-01-13T11:31:45.788Z	177.681062	-38.648667	1.524821	11.796875	M	NaN	...
2	2018p033712	NaN	2018-01-13T11:22:13.324Z	2018-01-13T11:23:58.188Z	175.578970	-39.188565	0.198787	5.468750	M	NaN	...
3	2018p033711	NaN	2018-01-13T11:21:17.372Z	2018-01-13T12:06:42.303Z	173.748415	-42.052923	2.610453	20.468750	M	NaN	...
4	2018p033704	NaN	2018-01-13T11:17:55.805Z	2018-01-13T11:20:20.927Z	177.439327	-37.307541	2.372378	39.687500	M	NaN	...

A sample of the data after processing:

	longitude	latitude	depth	origintime_s	quakes
0	173.786363	-42.207544	25.156250	1515810721	0
1	177.681062	-38.648667	11.796875	1515810592	0
2	175.578970	-39.188565	5.468750	1515810133	0
3	173.748415	-42.052923	20.468750	1515810077	0
4	177.439327	-37.307541	39.687500	1515809875	0

4 http://www.geonet.org.nz/

Using the mpl_toolkits.basemap[5] module it is possible to plot points on a map. The following picture is a visual representation of all the earthquakes from the dataset. Red represents a magnitude of less than 5.0, while blue is greater. The picture clearly shows that there are a lot of earthquakes that occur all over the entire country. Even though the entire country has earthquake points of less then a magnitude of 5.0, points greater than 5.0 are more isolated and only seem to occur in certain areas.



Algorithms and Techniques

The classifier is a Random Forest Classifier, an ensemble method of classifying using multiple decision trees.

While there are many parameters that can be tuned for the Random Forest Classifier, the following parameters were focused on:

- n_estimators – How many trees will be used within the forest.
- max_features – How many features will be used in a single branch before a new one is created.
- max_depth – How many levels of the tree are created.
- min_samples_leaf – The lowest number of nodes before a new tree is created.
- oob_score – A cross-validation technique within the classifier.
- For repeatability a random state of “42” was also chosen.

The dataset is split into two sets: training and testing. The split is a 70:30 in favour of training. The data is then be processed using the Random Forest Classifier, making sure that over-fitting is not taking place. As random forest can be prone to over-fitting the oob_score is set to true so that the score from each tree does not bias another tree within the forest. Since the data set has many points, a lot of trees are also used.

5 https://matplotlib.org/basemap/api/basemap_api.html

Benchmark

The benchmark for the project is a Naive predictor. The Naive predictor is a basic model that assumes an event either always happens or doesn't.

For this benchmark, the model will always predict and earthquake with a magnitude of greater than 5.0: a "1". This means that there will be no true-negatives or false-negatives.

In order to calculate the naive model the formulas to be used are:
(TP = true-positives; TN=true-negatives; FP=false-positives; FN=false-negatives); F-beta=1.25

- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$
- Precision: $\frac{TP}{TP+FP}$
- Recall: $\frac{TP}{TP+FN}$
- F-beta: $(1+\beta^2)\times\frac{precision\times recall}{(\beta^2\times precision)+recall}$

In the case of this project:
Accuracy is: 3584/587333 = 0.0061
Precision is: 3584/(3584+583749) = 0.0061
Recall is: 3584/3584 = 1
F-beta = $(1+1.25^2)*(0.0061*1)/((1.25^2*0.0061)+1)$ = **0.01**

The random forest classifier has a benchmark score of 0.01 to exceed.

III. Methodology

Data Preprocessing

The dataset used for this project contains 21 features in total. As a lot of these features are records of how the earthquake information, such as magnitude and depth, were recoreded and how the location, longitude and latitude, were calculated, these features were removed.
The features that were removed are:

- publicid – This is the unique reference ID of earthquake.
- eventtype – For this project, all of these will be "earthquake".
- modificationtime – If the information to the data point was updated, this time was recorded.
- evaluationmethod, evaluationstatus, evaluationmode, originerror, depthtype and earthmodel – These refer to how the earthquake was evaluated and what computer techniques were used.
- usedphasecount, usedstationcount, minimumdistance and azimuthalgap – These refer to how the location of the earthquake is calculated by the stations.
- magnitudetype, magnitudeuncertainty and magnitudestationcount – These refer to the calculation of the magnitude by the stations.

The final features that were kept are:

- origintime – Time of when the earthquake took place – Transformed into seconds
- latitude – Latitude of the earthquake
- longitude – Longitude of the earthquake
- magnitude – Strength of the earthquake – Are transformed to represent values above and below a magnitude of 5.0 as 0(<5) and 1(≥5).
- depth – How far underground the earthquake was located (in km).

The dataset also contained some data entries that were not earthquakes. These were listed in the "eventtype" feature. Any event that was either assumed to be or known as an earthquake was removed.

Within the dataset, 2 features also needed to be converted, so that the data was able to be used. The 'origintime' feature uses the ISO_8610[6] naming convention. (28th January 2018 at 9:00am JST would read as "2018-01-28T18:04:00+09:00") As such, the random forest classifier is not able to interpret this because it is seen as a string. Using the datetime module in python, this number is parsed into second, at which point the random forest algorithm is able to use it. Also due to limitations of the conversion process, events before 1900 will be removed. The base for the seconds to be calculated is 1900/01/01 00:00:00.

The second value to be converted is the magnitude feature. All values 5.0 and greater are converted to a '1' and all values less than 5.0 are converted to a '0'.

As mentioned in the data exploration section, the data is heavily imbalanced towards there being no earthquakes above a magnitude of 5.0. Only a 5.0 earthquake occurring 0.6% of the time. In order to make sure that the algorithm trains effectively on the data, the data was up-sampled at a ratio of 1:1. This means that the final data in use to train and test on has an equal number of earthquakes greater and lesser than 5.0. The final dataset has 1,133,868 points

6 https://en.wikipedia.org/wiki/ISO_8601

Implementation

The project was split into 6 stages.

Stage 1:

The data was loaded into python and the data preprocessing was carried out on the original dataset. The preprocessing was as described in the data preprocessing section of this report and carried out using python. At this point, six events were removed from the dataset to be tested at the end. Within these six points, there are three points where a large, 5.0 or greater earthquake appears and three where it does not.

Stage 2:

This stage was to calculate the Naive predictor that will be the benchmark for this project. The calculations can be seen in the benchmark section of this report. The calculation was done both within python and by pen and paper, the results of both coincided to give an answer of 0.01.

Stage 3:

Here, the data was up-sampled. As mentioned in the data exploration section, the original dataset is heavily imbalanced, only 0.6% of the data has earthquakes with a magnitude of 5.0 or greater. The data here is up-sampled using a package within sklearn called resample. It was chosen that the new ratio of events to non-events would be a 1:1 ratio.

Stage 4:

In the fourth stage the data is split into labels and features, and then again within these two groups the data is split into testing and training sets. The split for the training and testing sets is 70:30 split with training being 70% of the upsampled data and 30% for testing.

Stage 5:

This stage trains and tests the Random Forest Classifier. The classifier was also fine tuned at this stage to make sure that it was giving an optimal f-beta score and not over-fitting.

Stage 6:

This is the final stage where the sample points removed within stage 1 were then inputted into the Random Forest Classifier to test the overall prediction of the classifier. Some random future samples were also added into the classifier to see what the future values result in.

Refinement

The first instance of the Random Forest Classifier to run was with only the “random_state” parameter set. All other values were left at their default values: n_estimators = 10, max_features = auto(sqrt(n_features)), oob_score = False, min_samples_leaf = 1, max_depth = “None”.

With these parameters the accuracy and f-beta score on the testing set were both scoring above 0.999. This is possibly due to the random forest over-fitting.

Due to the suspected over-fitting, the model was then trained using an increasing amount of n_estimators (10, 100, 500, 1000, 2000), with a max_features = “sqrt” and oob_score = True.

n_estimators	Training F-beta score	Testing F-beta score
10	0.999982	0.999504
100	1.000000	0.999491
500	1.000000	0.999504
1000	1.000000	0.999509
2000	1.000000	0.999511

Due to the limitations of the computer doing the testing, the max number of estimators could not be increased above 2000. Also, since there is very little change between 500, 1000 and 2000 n_estimators, the following calculations will be done with 500 estimators due to the lower training time.

Since the scores are still very high, it was decided to change the min_sample_leaf on the n_estimators 500, with a max_features = “sqrt”, and oob_score = True.

min_sample_leaf	Training F-beta score	Testing F-beta score
1	1.000000	0.999504
2	0.999892	0.999273
3	0.999672	0.999032
4	0.999476	0.998828

The max_depth parameter was also changed using n_estimators 500, a max_features = “sqrt”, min_sample_leaf = 2, and oob_score = True.

max_depth	Training F-beta score	Testing F-beta score
1	0.759717	0.760146
5	0.833002	0.832015
10	0.893241	0.892774
20	0.997424	0.996364
50	0.999892	0.999273

The final values for the Random Forest Classifier are:

- n_estimators = 500
- max_features = “sqrt”
- oob_score = True
- min_samples_leaf = 2
- max_depth = 50
- All other features were left as default

Final training f-beta score: 0.999886
Final testing f-beta score: 0.999266

IV. Results

Model Evaluation and Validation

The values in the refinement section were chosen on a trial and error basis. Since the data set is so large, the time take to trial many different refinements was very time consuming. The scores were also very high, so changes to the model would only net less than 0.0001 change in the score. So it is possible that there is still some over-fitting going on.

Once the refinement was completed, six points of data were used as a prediction test.

Samples after preprocessing:

	longitude	latitude	depth	origintime_s	quakes
0	166.787490	-46.365530	5.000000	3.448276e+09	0
1	174.548230	-41.285370	30.680700	2.736774e+09	0
2	175.287182	-41.204894	21.640625	3.673823e+09	0
0	178.009990	-36.710000	12.000000	1.773488e+09	1
1	166.770950	-45.383360	20.143100	3.270459e+09	1
2	176.550000	-37.230000	348.000000	2.083324e+09	1

Where the ‘quakes’ column represents a earthquake greater than a magnitude 5.0 as a ‘1’ and ‘0’ to represent less than a 5.0.

Also a new data point that was not recored in the original dataset was added. This earthquake occurred on 2018/01/27 at 15:30:31 NZDT. It was approximately two weeks after the original dataset was downloaded and is a completely new point of data. This point is to test if the classifier can predict new future points.
The future point after preprocessing:

	longitude	latitude	depth	origintime_s	quakes
0	173.974579	-40.340115	123.553619	3.726009e+09	1

Using the above samples and the “future” point, the model was tested. The results of predict() and predict_proba() are as follows:

Sample number	Predicted value	Probability of predicted value	Actual value
Sample-1	0	100%	0
Sample-2	0	100%	0
Sample-3	0	100%	0
Sample-4	1	99.91%	1
Sample-5	1	99.27%	1
Sample-6	1	99.62%	1
Future point	0	100%	1

The classifier was able correctly assign the values that were originally within the main dataset. Although, with the new “future” point it was not able to correctly assign it.

Justification

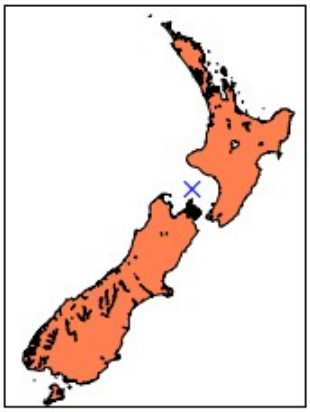
The original Naive predictor benchmark gave an f-beta score of 0.01. In contrast to that, the tuned Random Forest Classifier was able to achieve an f-beta testing score of 0.999273. Since the classifier's f-beta score is very close to being perfect, it can be assumed that almost all outputs are correct. Using the table in the evaluation section where the sample points are tested, six out of seven points are correct. Which is about 85% correct.

The reasoning behind the single point being incorrect could be due to the gap in data between the most recent point in the main dataset and the new data point.

V. Conclusion

Free-Form Visualization

The “future” point was located within the ocean close to the capital city of Wellington. While this time, the earthquake was located rather deep at 123.6km. Since the point was located rather deep, there was no tsunami. A more shallow earthquake could have caused more destruction. With accurate prediction destruction could be averted.



The location of the “future” point overlaid on a map of New Zealand.

Reflection

A quick summary of this project is:

1) An initial idea and problem was though up.
2) Dataset(s) were searched for – the main requirement for these being free, and publicly available.
3) The data was cleaned up and processed.
4) A benchmark for the project was calculated.
5) The classifier was chosen and an initial run was conducted.
6) The classifier was tuned and re-run with the new parameters.
7) A new, real data event was searched for to trial with the classifier.

The most surprising point about this project was how high the classifier f-beta score did on the first run with no tuning of the parameters. I was expecting it to be a lot lower. After tuning the classifier I found that it had been over-fitting the data a little, but the values were still very high. I found that the most difficult point was fine tuning the classifier. Because of the large dataset and there not being much other information on the Internet about datasets with a very high number of data points with few features, it was difficult to tell if the model was working as expected.

I am still not sure if the classifier is able to predict future events. There still needs to be a lot more work done on this to see if there is any viability in it, but because of the large dataset and limitations of my computer the calculation takes a lot of time.

Improvement

There are two points that I think could be improved on for this project. The first point would be to have the data constantly being fed into the classifier. The GeoNet website has the option to connect through an API. I think if the classier was able to use this it might be able to better predict future points as it is always learning.

A second improvement would be to create more random simulated data. An example of this could be using historic data to create new data such as longitude, latitude and depth and then increment the time to see if there are any predictions.