Team Earthquick Project Summary

Authors: Tingkai Zhang, Dirk Cornelis Haupt, Osman Hajiyev

We are team Earthquick and we will be presenting our new and advanced machine learning approach to earthquake prediction.

Earthquakes:
Damages caused by earthquakes are enormous. There are on average 1700 earthquakes with a magnitude of 5 or more (M5+). On average, annually 50.101 lives are lost due to earthquakes (2000-2015). Hundreds of thousands of people lose their home or property and tens of billions of dollars are lost every year in all kinds of damages because of earthquakes. Earthquakes are capable of causing environmental damage as well (nuclear disasters for example). Moreover, earthquakes cause other very deadly natural disasters, like tsunamis and avalanches.

State of the earthquake prediction:
A lot of these losses and damages could have been mitigated or potentially even mostly eliminated with a better earthquake prediction mechanism, however so far, the state of the earthquake prediction has been disappointing and completely unreliable. Successful predictions of large earthquakes so far have not occurred.

Solution:
However, in the last 3 years the rise of machine learning and more powerful cloud computing opened up a lot more opportunities at solving exactly this kind of prediction problem. Using ML we decided to tackle the earthquake prediction challenge. We used amazing technologies available through Microsoft Azure and for our data we used artificial lab earthquake datasets that are publicly available through Los Alamos National Laboratory and as a result ended up doing decent and successful predictions of earthquake timings! We will explain in more detail now:

DevOps
Data is stored in Azure Storage gen 2. Azure databricks is used to explore the data and do the preliminary analysis on the model. We trained Machine learning models using 'Microsoft Machine Learning for Apache Spark'(MMLSpark) in Databricks. Key Vault-backed secret scope is created in Databricks to enhance the security. In databricks, the code will read data from Azure blob storage. Azure Machine learning Service is used to manage the model and monitor the experiments. Azure KeyVault is used to store secret variables. We use Azure DevOps to continuously build and deploy the model.
In the build phase, a docker image is created. The build artifact is served as the input of the release pipeline. In release phase, the docker image is deployed on Azure Container Instance, which serves as http endpoint. We use Azure Container Instance (ACI) to test the project's functionality. We create the production environment and deploy the container on Azure Kubernetes Service cluster.
During the whole project, we use git as our version control system.

Engineering
We first did some research on the problem background, including earthquake science and earthquake lab experiments. Then we did data exploration to understand the statistics description of the data. We studied the domain knowledge of earthquake prediction by reading the research papers. In feature engineering step, we applied signal processing techniques (like calculating Mel Frequency

Cepstral Coefficient) to generate valuable features. We tried different machine learning models like XGBoost, random forest and neural network.
We select LightGBM as the final model due to its best performance and its optimization on Spark

Web App
We created a Flask web application as our Minimal Viable Product (MVP) that seismologists and experts can use to interact with our model and prediction results. Users can use the app to upload their test data which is first pre-processed and then a POST request is sent to the docker image http endpoint that we deployed via Azure Databricks. Data is retrieved and the raw acoustic data along with the predicted time to failure is displayed using an interactive visualization framework known as Plotly.
Our MVP is already deployed to Heroku and can be viewed here: https://earthquick.herokuapp.com/. Note though that the deployed app does not currently work at scale as the pre-processing step (potentially involving millions of data-points) has not yet been parallelized. In the next phase we will use Azure Batch to run these large parallel jobs. We further also plan to migrate from the Plotly data visualization framework to Dash in the next phase as it is known to integrate better with Flask.
We also allow the user to update and add to the training dataset. We have a separate upload function where the user appends labeled data to the training set using Azure Blob Storage and AppendBlobService. The model is then rebuilt in Databricks. This allows for all users of the app to contribute their data and improve the model for everyone.

Potential Impact:
This is just the beginning and it is only a minimum viable product. There is still huge room for improvement for our earthquake prediction model and app, given more time we would be able to apply more advanced models and provide more variables which our model can potentially make use of in order to make even more accurate predictions. Potential impact could be enormous: tens of thousands of lives could be saved annually, financial and property losses which are currently at tens of billions of dollars annually could be minimized. Hundreds of thousands could be saved from losing their home. Our solution will mitigate the damage partially caused by climate and environmental change (UN SDG Goal 13). It will potentially dramatically improve on SDG goals 3 (Health and well-being) and 11 (Sustainable cities). The demand for this product is big and it would mostly be demanded by governments and corporations seeking to minimize all kinds of damages. Insurance companies would especially be interested in our work. At this stage the app is mostly targeted at seismologists, scientists and experts who will be able to interact with our model and the app and predict earthquake timings with a decent success chance thus minimizing financial and human losses.

With the help from the latest tools from Microsoft our solution is a big improvement on historical earthquake predictions and can potentially save hundreds of thousands of lives and hundreds of billions in infrastructure and property damages.