Volcanic Eruption Prediction

Shani Klein – [redacted]

Elinor Peer [redacted]

Table of Contents

Introduction

Anticipating volcanic eruption could make a big impact. Just one unforeseen eruption can result in tens of thousands of lives lost. If we could reliably predict when a volcano will next erupt, evacuations could be more timely and the damage mitigated.

Goal+ Approaches

Our main goal in this competition was finding a suitable and applicable model in order to predict when the next volcano eruption will occur. We used different Machine Learning methods learned in class to analyze a large geophysical dataset collected by sensors deployed on active volcanoes .

We designed and trained a machine learning model that predicts the time of eruption , based on our explanatory variables. We wrote our code in Python using sklearn, numpy, pandas, seaborn, scipy, matplotlib libraries

Research questions and challenges

Which features can we extract that will be most relevant and meaningful data.

What are the relationships between those features and the eruption time.

Should we take all the features we extracted or apply Feature Selection?

Which set of statistical processes (i.e model) will be best for estimating the relationships between the features and time to eruption.

Should we apply pre-processing foe the features ? for example applying Dimension reduction or Standardize the features.

More research questions and challenges

What is the best way to deal with outliers?

How can we handle missing data and poor quality of data?

How to cross validate efficiently over small data sets?

How to avoid overfitting and underfitting while training the data.

Which measure of errors should we take to estimate the performance of our model?

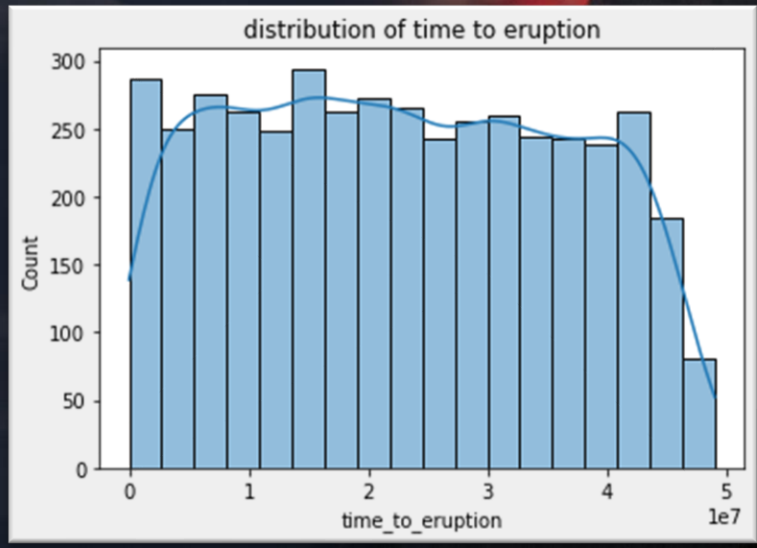
Data Overview and Exploration

We got data from Kaggle competition. The competition provides us with readings from several seismic sensors around a volcano , and how long it will take until the next eruption. The data represent a classic signal processing setup that has resisted traditional methods.

The dataset was divided into two data sets: training set which contains 4520 files and test set that contains 4431 filed . The problem is a supervised learning, hence each file is has “time to eruption” label column .

In order to observe our data shape and values, we visualized the train dataset by doing histogram plot to see the distribution counts of the time to eruption.

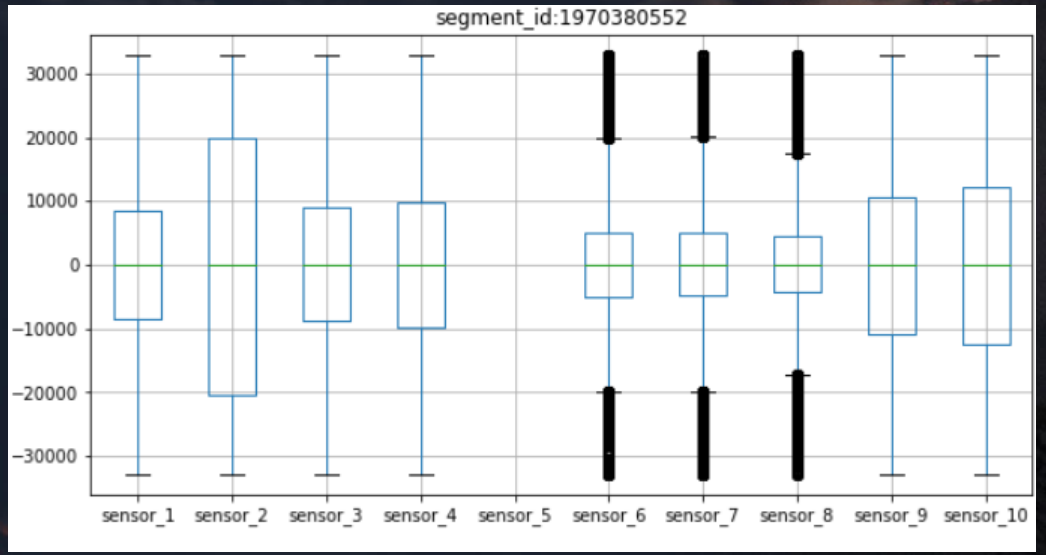
From the distribution represented we can tell That the eruption time vary from zero to $5 \cdot 10^7$. In addition only few segments has time_to_eruption of $5 \cdot 10^7$, That may indicate about an outlier, we will observe it later on .



Data Overview and Exploration:

Next, we explored how the train files look like. Each train file contained data from 10 different sensors. We randomly chose 3 files and depicting them by using boxplot.

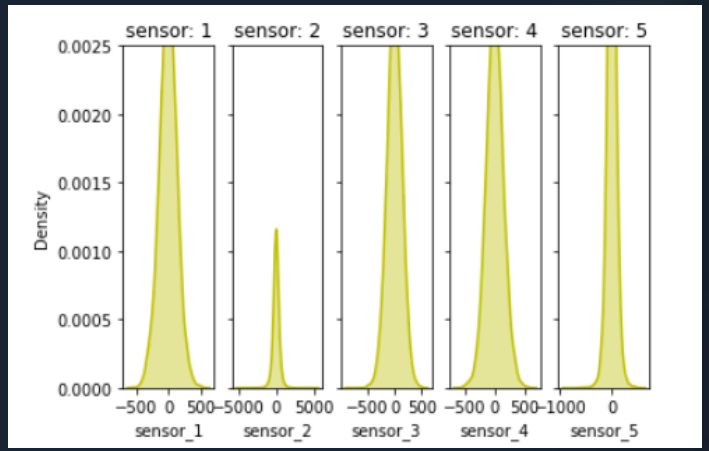
For example, observing the following boxplot we can tell that the median of each sensor is zero.
In addition, are some missing values , For example in here , sensor 2 has missing data. The missing values were handled in the feature extraction step.

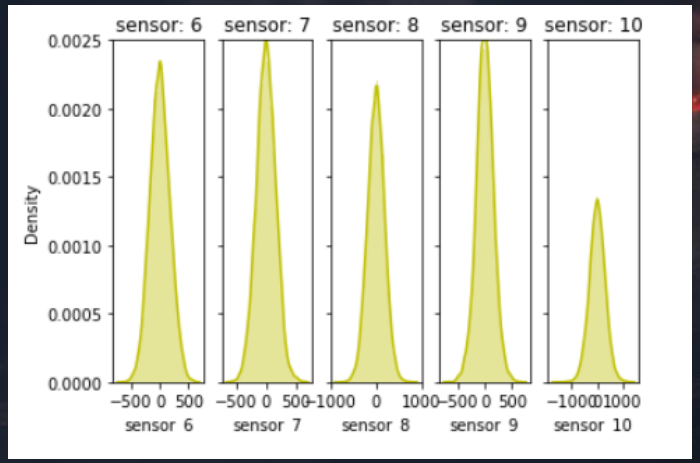


Data Overview and exploration:

Next, we used density plot in order to examine where the peaks are located, which display where the values are concentrated over the interval.

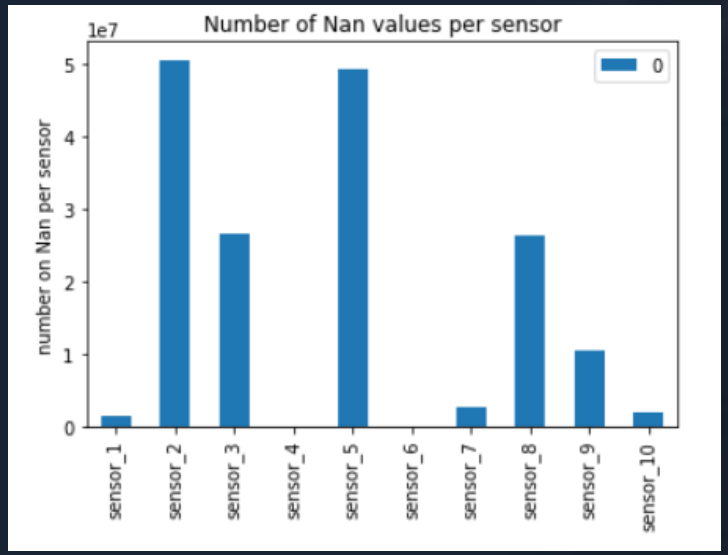
We can tell all the sensors has peaks around zero .

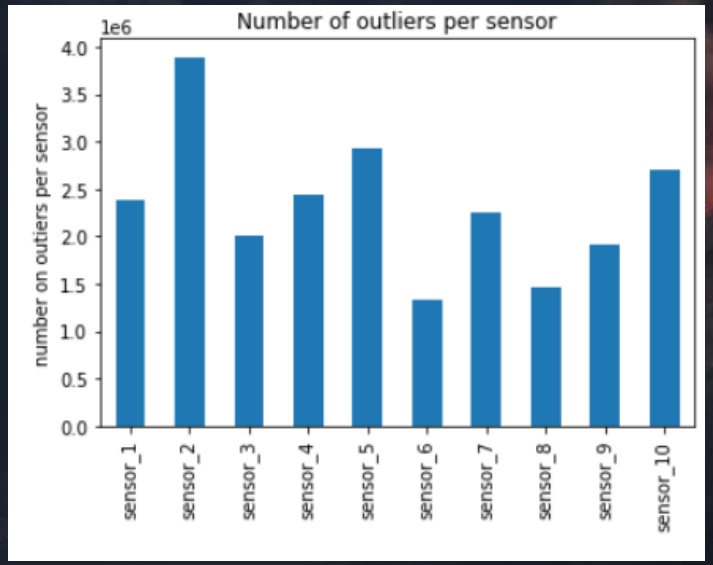




Identify missing values and outliers

- First of all , we want to identify null value and outliers. We observed few methods to decide which point will be considered an outlier , we will discuss it later on .





Identify missing values and outliers

	number of null	number of outliures
sensor 1	1545802	2383048
sensor 2	50569629	3891346
sensor 3	26499171	2004380
sensor 4	6210	2435301
sensor 5	49241506	2920901
sensor 6	88904	1340445
sensor 7	2546627	2248834
sensor 8	26465572	1461074
sensor 9	10520113	1911212
sensor 10	1895973	2699513

We calculate the percentages of the outliers from the data in order to get some sense about them and got that the outliers is only 0.0876% from the data

Dealing missing values

We observe 2 option to deal with missing data:

1. **Deletion**: we considered to delete an entire record if a single value is missing . This solution may be simple, but we encounter few problems:
   a. It reduce the sample size significantly , and we got insufficient data .
   b. We delete entire row even if only one value is missing , and the leads us to miss some essential information.

2. **Imputation :** we decided to replace the missing value with some value we inferred from the data . The options we observe:
   a. Replace the missing value with the mean of the column.
   b. Fill with zero .
   c. Sort records according to some criteria , and for each missing value, use immediately prior available value.

   We decided to do imputation , for reasons of simplicity we chose to replace missing value with the mean of the column.

Dealing with Outliers

An outlier is a data point that differ significantly from other data points in the same data set.

To deal with outlier we follow 2 steps:

Identifying the outliers- we observed 2 options to detect the outliers

1. **Interquartile Range**– the interquartile range (IQR), is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles,

   $IQR = Q3 - Q1$.

   We tried different measurement and decided to consider each dot that below $Q_1 - 1.5$ or above $Q_3 + 1.5$ as an outlier.

2. Distance from fitted line – Another approach we tried was to do additional regression analysis and fit the data to some line , then outlier will be point that do not fit into the line .

We decided to choose the first method , and decided **to remove all outliers .**

Feature Selection

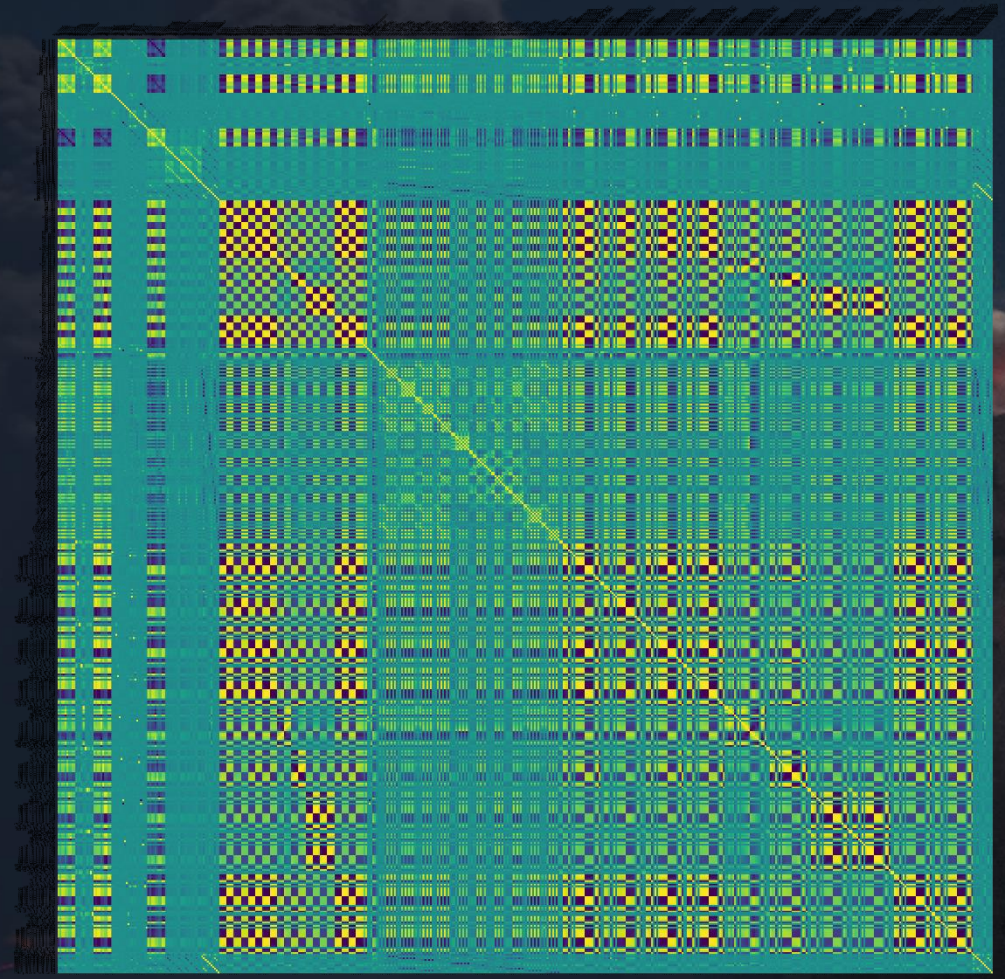
Feature selection Is a method when we use a subset of X variables to train our model . Feature selection techniques are useful for several reasons such as simplification of models, shorter training times, to avoid the curse of dimensionality, enhanced generalization by reducing overfitting. In addition , we want to use feature selection to filter out noise and to extract the relevant information from the given data .

We observe two methods to apply Feature Selection :

- One method is to check the correlation of each feature , features with low correlation were removed. The term low correlation was decided according to a given threshold.
  We tried different values of thresholds and got the best result with threshold of 0.01.
- Using Lasso regression- we will explain this method in the next slide

Correlation Matrix

- We plot the correlation matrix, there are a lot of features so it is hard to identify each feature, but we can tell that there some features with high correlation one another and some with low correlation.



Lasso Regression for Feature Selection

As we learnt in class , linear regression tries to find the best fit line that minimize the error (in our case is MAE) . When using Lasso regression , a new term is added to the equation and instead of find A,B that minimize y=Ax+B we need to find A,B that minimize :

$$y = Ax + b + \alpha(|A| + |B|)$$

When α is hyperparameter .

We basically add some penalty to the model, a kind of regularization .

The use of this penalty forces the regression model to keep things simple (A and B cannot be really big numbers) , hence using Lasso regression can help mitigation over-fitting on the train data.

When α is really big that force small coefficient to zero , what allows Lasso Regression to select relevant features. Tweaking the value of α allows us to perform model selection by selection only relevant feature with large coefficient .

In our code – we apply Lasso Regression with Kfold=5 and MAE as the scoring to find best fit for the data , then – we use the lasso as a Feature selection by selection only the features which their coefficient in not zero .

Feature scaling

Since the range of values of the raw data varies widely, the classifier may not work properly without normalization .

Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final classification.

We dealt this issue by using StandardScaler of sklearn that standardize features by removing the mean and scaling to unit variance.

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set using the fit method. Mean and standard deviation are then stored to be used on later data using transform.

Dimension reduction – PCA

- As we learnt in class PCA is a simple, non-parametric method of extracting relevant information from confusing data sets, and it provides a roadmap for how to reduce a complex data set to a lower dimension . First we tried to use PCA to perform a change of basis on the data .
- We took all components that explained over 99% of the variance, eigenvectors vectors with zero.
- As we said in the previews slides we used feature scaling to normalize the range of independent features of data. Scaling is important while performing PCA which tries to get the features with maximum variance, and the variance is high for high magnitude features and skews the PCA towards high magnitude features.

Dimension reduction –Kernel PCA

- PCA tries to find a low-dimensional linear subspace that the data are confined to. But it might be that the data are confined to low-dimensional nonlinear subspace.
- Hence one of the improvement we did was move from performing PCA to perform Kernel PCA . Kernel PCA is Non-linear dimensionality reduction through the use of kernels , It does so by mapping the data into a higher-dimensional space.
- Is it assuming a much higher dimensional space so it is able to reveal the nonlinear relations in the data, compared to PCA.
- We tried different kinds of kernels and found that the linear kernel was the best fit for us .

Choosing and Training a model

The models we tried was:

- Linear regression
- Ridge regression
- Kernel Ridge Regression using different kernels.

We got the best results using Kernel Ridge using RBF as the kernel.

In the next slides we will explain all the models we tried and the problems and difficulties we encountered.

Linear Regression and Multicollinearity

The first model we tried was linear regression which we learnt in class.

Few problems we encounter :

1. First, Linear Regression is perform best to minimize the MSE between the observed targets in the dataset, and the targets predicted by the linear approximation. The competition scores by MAE ,thus we can get bad result in linear regression but in the competition it will “punish” us more softly. One of the problems was that we felt like the score we get on the linear regression wasn’t enough informative

2. A key goal of regression analysis is to isolate the relationship between each independent variable and the dependent variable when using linear regression. Linear regression has a problem called Multicollinearity which occurs when independent variables in a regression model are correlated. This correlation is a problem because independent variables should be *independent*. If the degree of correlation between variables is high enough, it can cause problems when you fit the model and interpret the results

Ridge Regression

- Ridge Regression is particularly useful to mitigate the problem of multicollinearity in linear regression, which commonly occurs in models with large numbers of parameters.
- Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value.
- In Ridge Regression, the ordinary least squares loss function is augmented in such a way that we not only minimize the sum of squared residuals but also penalize the size of parameter estimates, in order to shrink them towards zero
- By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable

Kernel Ridge Classification

- Kernel ridge regression combines Ridge regression and classification (linear least squares with l2-norm regularization) with the kernel trick. It thus learns a linear function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a non-linear function in the original space .

- We tried different types of kernels such as linear kernel , Laplacian, RBF and gaussian .

- The best result we got was with RBF kernel : $K(x, x') = \exp\left(-\frac{|x-x'|^2}{2\sigma^2}\right)$ when σ is a free variable.

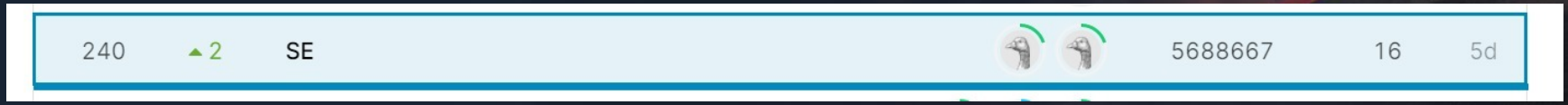
Cross validation + Results

in order to utilize our data better we used K fold cross validation method and split the data to 7 folds. Given relatively small data sets this method provide us more metrics to draw conclusion both about our algorithm and our data.

The results we got :

```
Train MAE: 638934.373
Validation MAE: 4109516.634
Test MAE: 4360450.179
```

And the result in the competition :



References

- https://www.kaggle.com/amanooo/ingv-volcanic-basic-solution-stft#INGV-Volcanic-:-Basic_solution-(STFT)

- https://www.kaggle.com/jesperdramsch/introduction-to-volcanology-seismograms-and-lgbm

- Kernel ridge explanation – https://www.ics.uci.edu/~welling/classnotes/papers_class/Kernel-Ridge.pdf