<u>**Earthquake Prediction Using Machine Learning: 1. Gradient Boosting (CatboostRegressor)**</u>

<u>**2. Support Vector Machine (SVM)**</u>

Lecture Content

- Earthquake prediction background & helpful resources
- Step 1 - Installing dependencies
- Step 2 - Importing dataset
- Step 3 - Exploratory data analysis
- Step 4 - Feature engineering (statistical features added)
- Step 5 - Implement "Catboost" model
- Step 6 - Implement Support Vector Machine + Radial Basis Function model

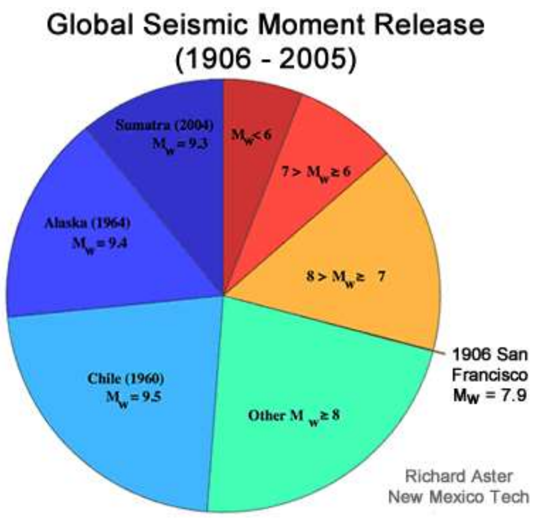
EarthQuake Prediction Background

- Predicting earthquakes has long been thought to be near-impossible.
- Being able to predict earthquakes could allow us to better protect human life and property.

Research Prediction Competition
LANL Earthquake Prediction
Can you predict upcoming laboratory earthquakes?
$50,000
Prize Money
Los Alamos National Laboratory · 996 teams · 4 months to go (4 months to go until merger deadline)
Overview
Data
Kernels
Discussion
Leaderboard
Rules
Team
My Submissions
Submit Predictions

There are many ways to compute the magnitude of an earthquake!

- A0 is the seismometer reading produced by an Earthquake of standard size (i.e., a calibration earthquake). Generally A0 is 0.001 mm.
- This equation assumes that a distance of 100 km separates the seismometer and the



- epicntre

- Given seismic signals we are asked to predict the time until the onset of laboratory earthquakes
- In the lab, 2 plates are put under pressure, resulting in shear stress
- The training data is a single sequence of signal and seems to come from one experiment alone.
- The test data consists of several different sequences, called segments, that may correspond to different experiments.
- For each test data segment with its corresponding seg_id we are asked to predict it's single time until the lab earthquake takes place.

Also, shoutout to Anton Loss. He turned this data into sound, then compared it with dubstep.

Anton's work here

Anton's hypothesis: perhaps just like a dubstep song, we can hear the build-up of an earthquake before the drop

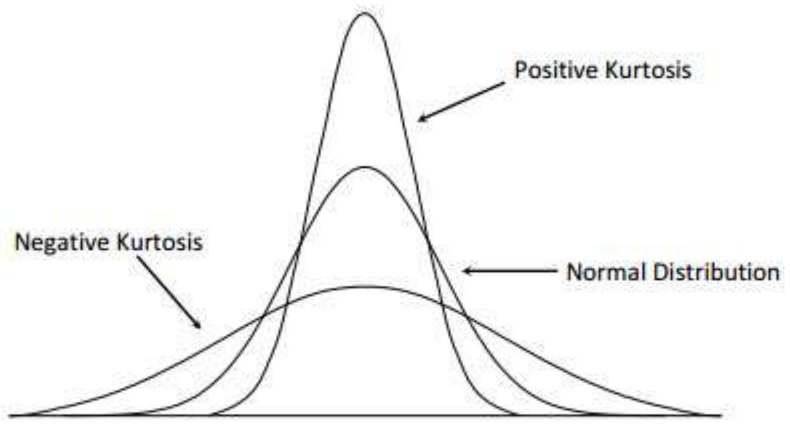
1. The data looks like sound waves, it oscillates! Thus, he converted it into an audio file.
2. He then denoised the audio file
3. Next, he animated the wave
4. Finally, he compared it to a dubstep song

Mean of Grouped Data:

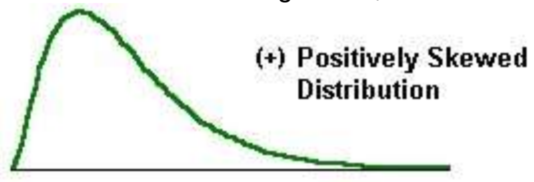
$$\bar{x} = \frac{\sum fx}{n}$$

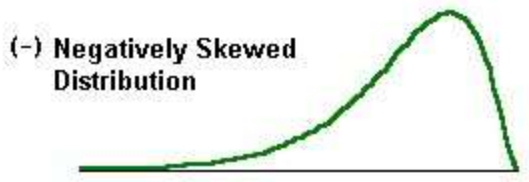
where:

$\bar{x}$	=	*mean*
f	=	*frequency of each class*
x	=	*mid-interval value of each class*
n	=	*total frequency*
$\sum fx$	=	*sum of the producst of mid − interval values and their corresponding frequency*

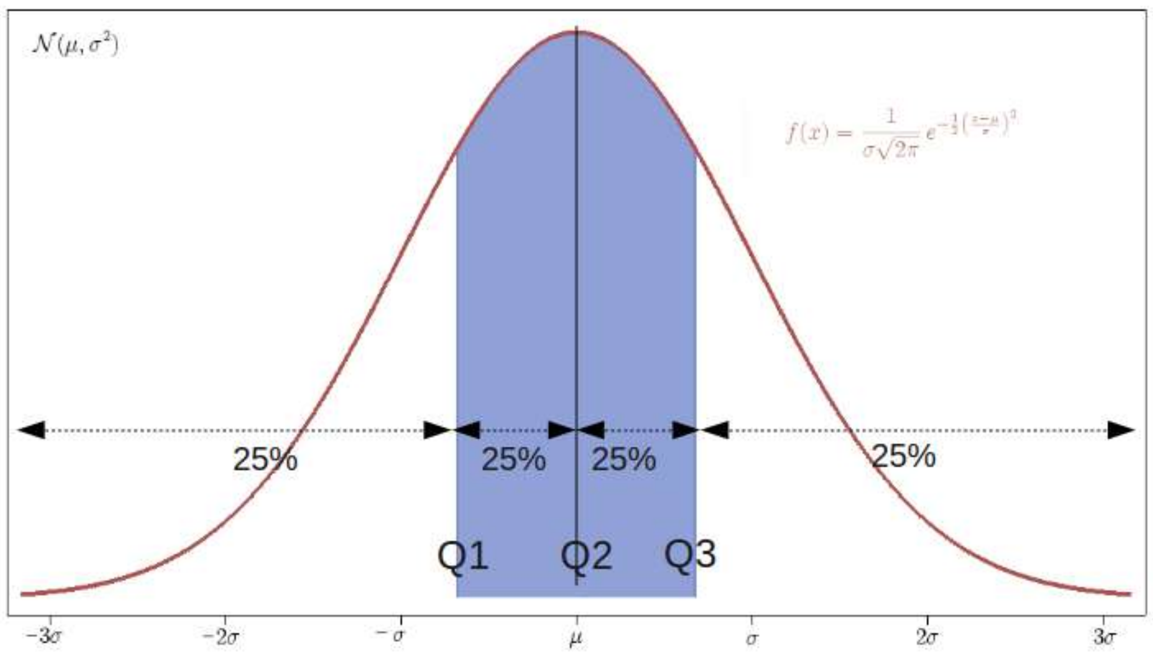


- Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution.
- Data sets with high kurtosis tend to have heavy tails, or outliers. Data sets with low kurtosis tend to have light tails, or lack of outliers





- skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive or negative, or undefined.

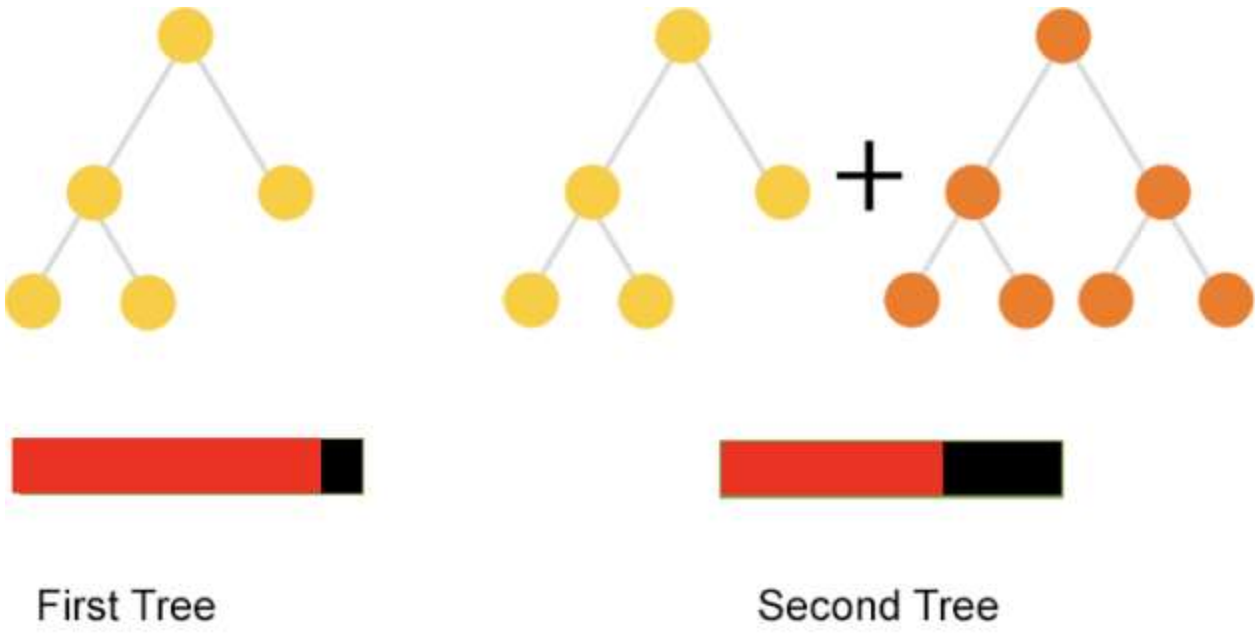


- Quantiles are cut points dividing the range of a probability distribution into continuous intervals with equal probabilities, or dividing the observations in a sample in the same way.

In [0]:

Step 5 - Implement Catboost Model

- Yandex is Russian Google
- Yandex uses "gradient boosting" a lot to power their services (music streaming, search, everything really)
- Gradient boosting on decision trees is a form of machine learning that works by progressively training more complex models to maximize the accuracy of predictions.
- It's particularly useful for predictive models that analyze ordered (continuous) data and categorical data.
- It's one of the most efficient ways to build ensemble models. The combination of gradient boosting with decision trees provides state-of-the-art results in many applications with structured data.
- On the first iteration, the algorithm learns the first tree to reduce the training error, shown on left-hand image in figure 1.
- This model usually has a significant error; it’s not a good idea to build very big trees in boosting since they overfit the data.
- The right-hand image in figure 1 shows the second iteration, in which the algorithm learns one more tree to reduce the error made by the first tree.
- The algorithm repeats this procedure until it builds a decent quality model



- Gradient Boosting is a way to implement this idea for any continuous objective function.

Each step of Gradient Boosting combines two steps:

- Step 1 - Computing gradients of the loss function we want to optimize for each input object
- Step 2 - Learning the decision tree which predicts gradients of the loss function

ELI5 Time

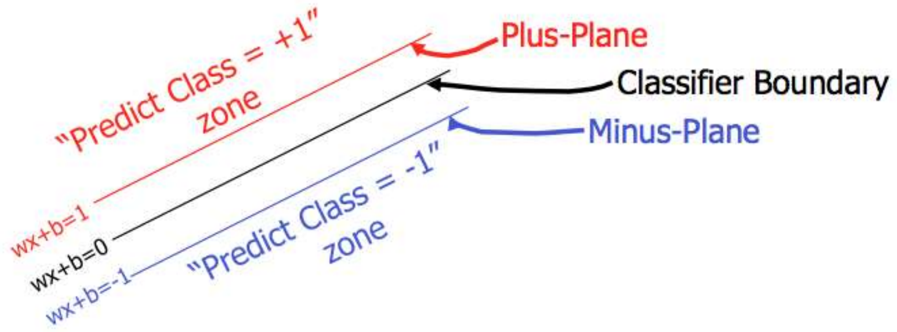
- Step 1 - We first model data with simple models and analyze data for errors.
- Step 2 - These errors signify data points that are difficult to fit by a simple model.
- Step 3 - Then for later models, we particularly focus on those hard to fit data to get them right.
- Step 4 - In the end, we combine all the predictors by giving some weights to each predictor.

Step 6 - Implement Support Vector Machine + Radial Basis Function Kernel

you might be thinking WTF YOOOOoooo. thats ok, breathe. here we go.

Need to learn a nonlinear decision boundary? Grab a Kernel

- A very simple and intuitive way of thinking about kernels (at least for SVMs) is a similarity function.
- Given two objects, the kernel outputs some similarity score. The objects can be anything starting from two integers, two real valued vectors, trees whatever provided that the kernel function knows how to compare them.
- The arguably simplest example is the linear kernel, also called dot-product. Given two vectors, the similarity is the length of the projection of one vector on another.
- Another interesting kernel examples is Gaussian kernel. Given two vectors, the similarity will diminish with the radius of σ. The distance between two objects is "reweighted" by this radius parameter.
- The success of learning with kernels (again, at least for SVMs), very strongly depends on the choice of kernel. You can see a kernel as a compact representation of the knowledge about your classification problem. It is very often problem specific.



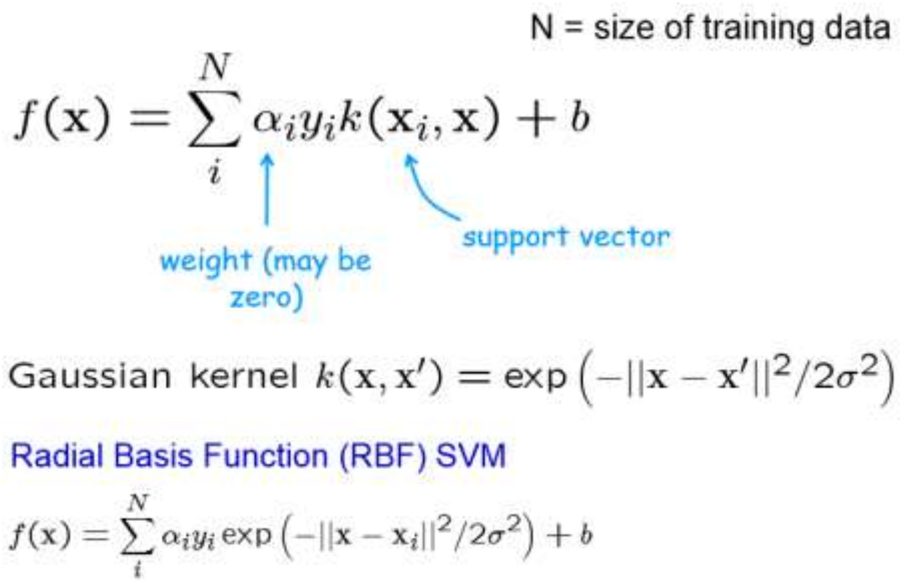
- Plus-plane = $\{ \mathbf{x} : \mathbf{w} \,.\, \mathbf{x} + b = +1 \}$
- Minus-plane = $\{ \mathbf{x} : \mathbf{w} \,.\, \mathbf{x} + b = -1 \}$

Classify as..	+1	if	$\mathbf{w} \,.\, \mathbf{x} + b >= 1$
	-1	if	$\mathbf{w} \,.\, \mathbf{x} + b <= -1$
	Universe explodes	if	$-1 < \mathbf{w} \,.\, \mathbf{x} + b < 1$

Support vector machines

- Common kernel functions for SVM

– linear	$k(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$
– polynomial	$k(\mathbf{x}_1, \mathbf{x}_2) = (\gamma\, \mathbf{x}_1 \cdot \mathbf{x}_2 + c)^d$
– Gaussian or radial basis	$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\gamma \lVert \mathbf{x}_1 - \mathbf{x}_2 \rVert^2\right)$
– sigmoid	$k(\mathbf{x}_1, \mathbf{x}_2) = \tanh(\gamma\, \mathbf{x}_1 \cdot \mathbf{x}_2 + c)$



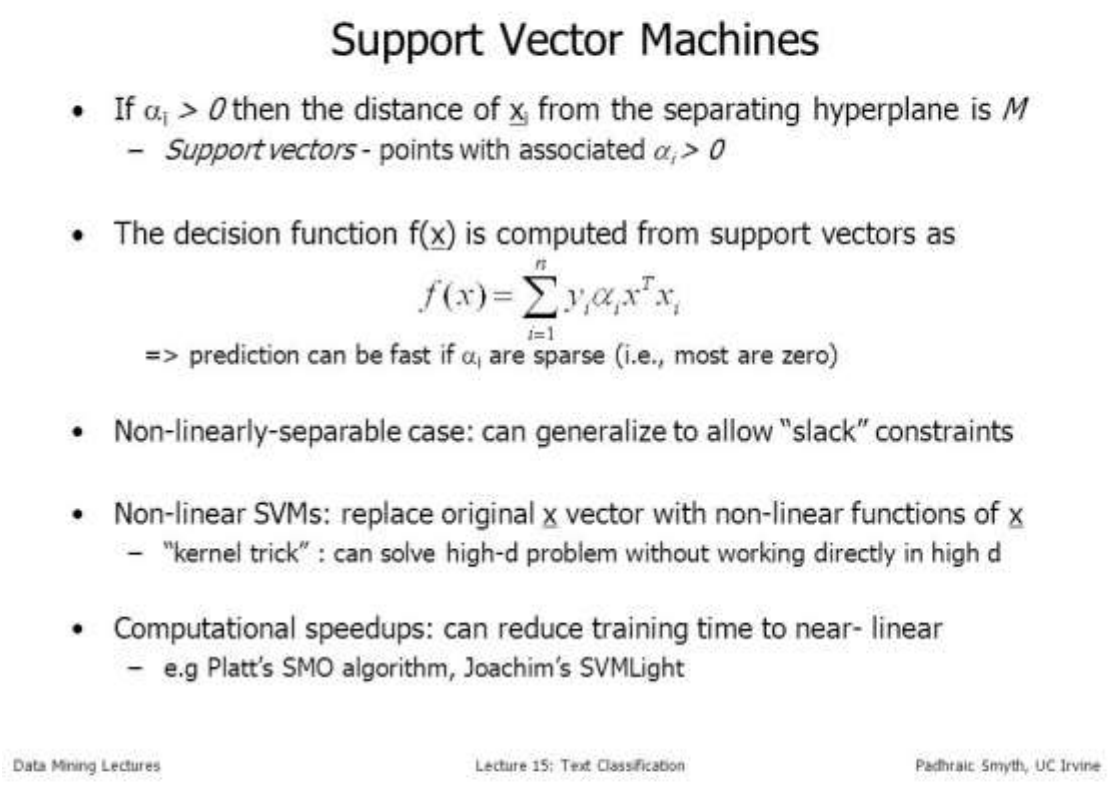
N = size of training data

$$f(\mathbf{x}) = \sum_{i}^{N} \underset{\text{weight (may be zero)}}{\alpha_i} y_i k(\underset{\text{support vector}}{\mathbf{x}_i}, \mathbf{x}) + b$$

Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-||\mathbf{x} - \mathbf{x}'||^2 / 2\sigma^2\right)$

Radial Basis Function (RBF) SVM

$$f(\mathbf{x}) = \sum_{i}^{N} \alpha_i y_i \exp\left(-||\mathbf{x} - \mathbf{x}_i||^2 / 2\sigma^2\right) + b$$



Support Vector Machines

- If $\alpha_i > 0$ then the distance of $\underline{x}_i$ from the separating hyperplane is M
  - *Support vectors* - points with associated $\alpha_i > 0$
- The decision function f($\underline{x}$) is computed from support vectors as

$$f(x) = \sum_{i=1}^{n} y_i \alpha_i x^T x_i$$

  => prediction can be fast if α_i are sparse (i.e., most are zero)
- Non-linearly-separable case: can generalize to allow "slack" constraints
- Non-linear SVMs: replace original $\underline{x}$ vector with non-linear functions of $\underline{x}$
  - "kernel trick" : can solve high-d problem without working directly in high d
- Computational speedups: can reduce training time to near- linear
  - e.g Platt's SMO algorithm, Joachim's SVMLight

Data Mining Lectures Lecture 15: Text Classification Padhraic Smyth, UC Irvine

grid search : pick a bunch of values of α -- (α1,α2,…), pick a bunch of values of β -- (β1,β2,…) and for each pair of values, evaluate the validation error function. Then pick the pair that gives the minimum value of the validation error function.

The pairs (α1,β1),(α1,β2),…,(α2,β1),(α2,β2),… when plotted in space look like a grid, hence the name.

In [56]:

Future Ideas

- Recurrent Networks
- Genetic Algorithms
- Ordinary Differential Equation Networks (YOOOOOO)