# EEG Seizure Detection Competition

BitTiger丨来自硅谷的终身学习平台

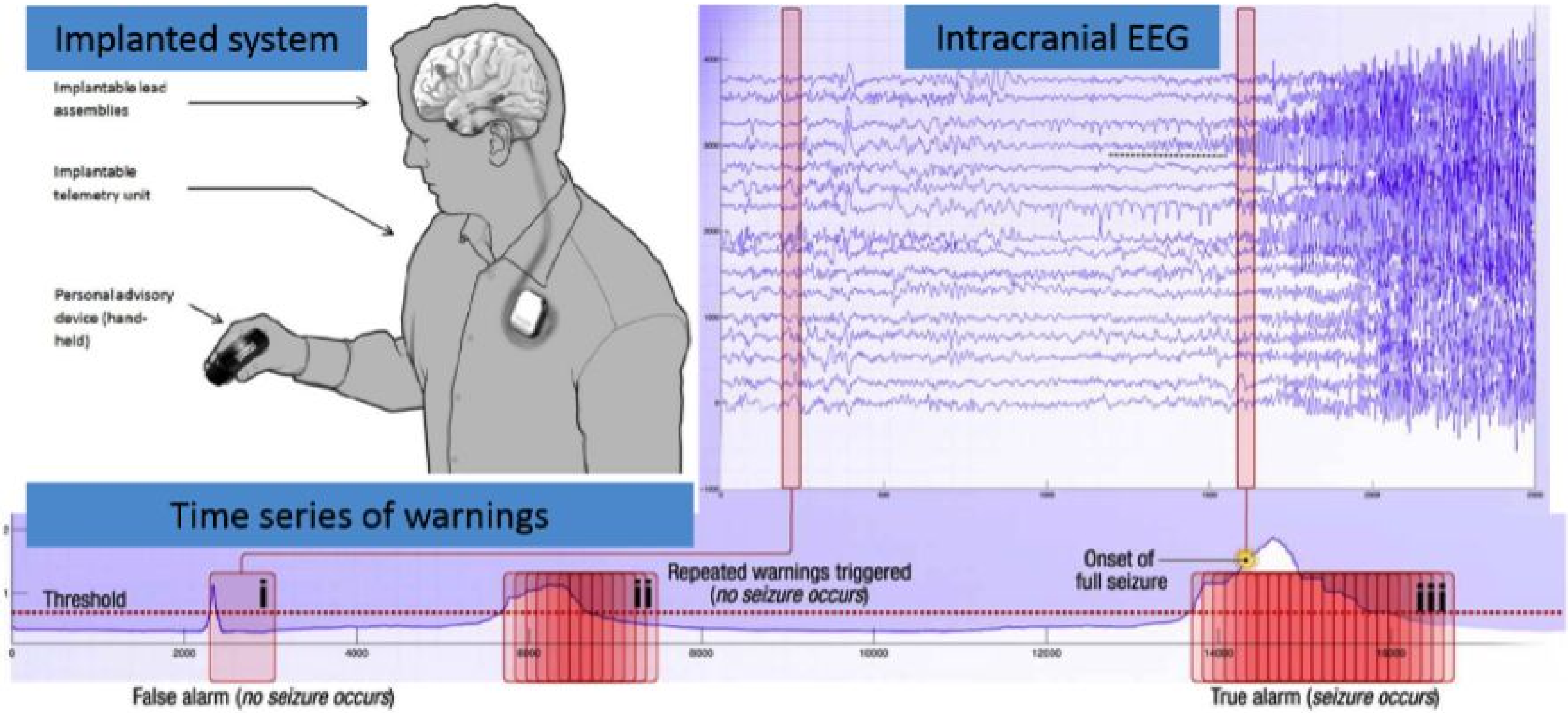
Rand Xie

Disclaimer Any opinions and code here are my own, and in no way reflect that of MathWorks

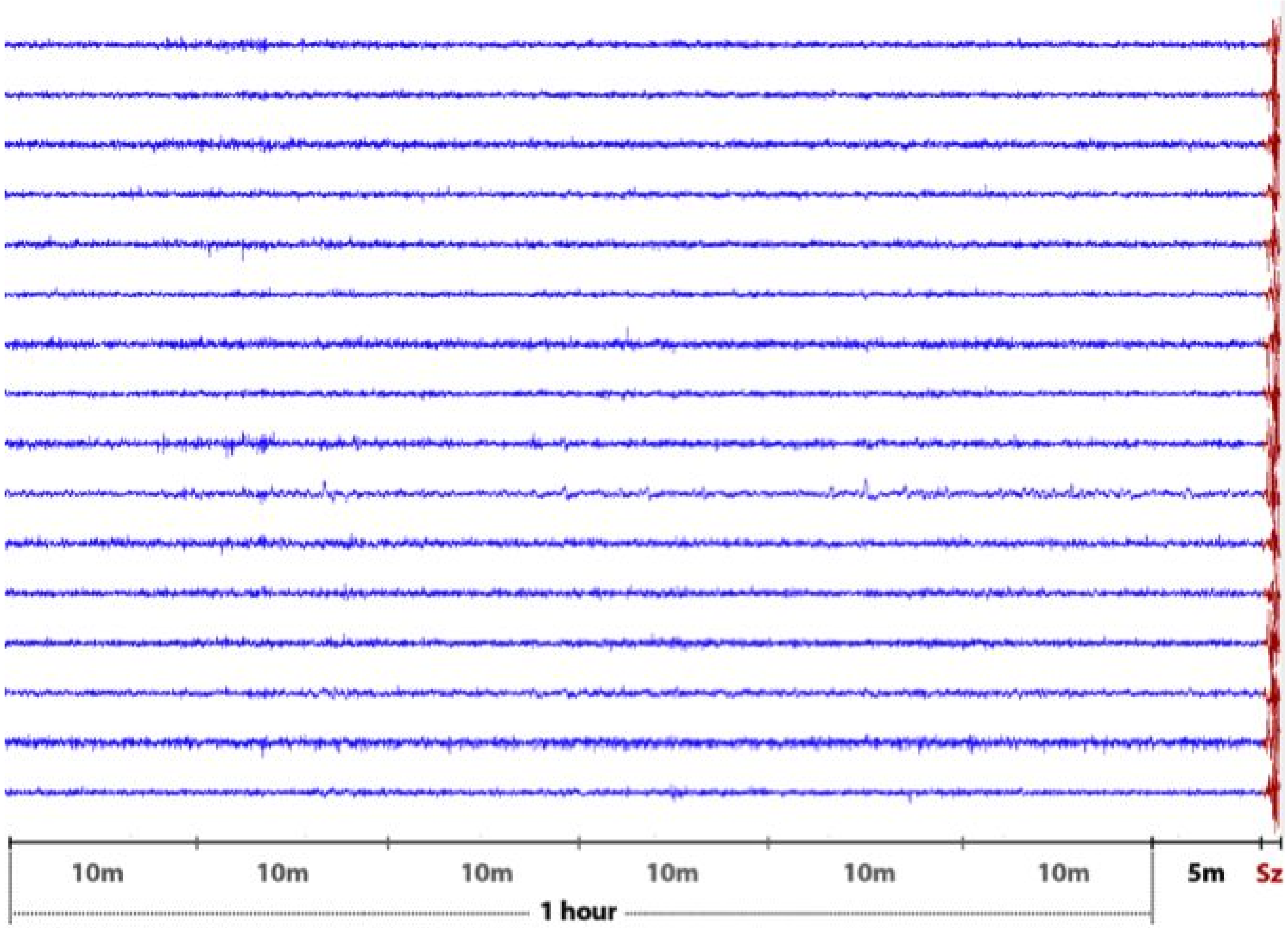
# Outline

- Introduction to EEG dataset and Kaggle community
- Time series feature extraction
- From loading data to submission
- Introduction to feature selection
- Introduction to parameter tuning

# Seizure Prediction



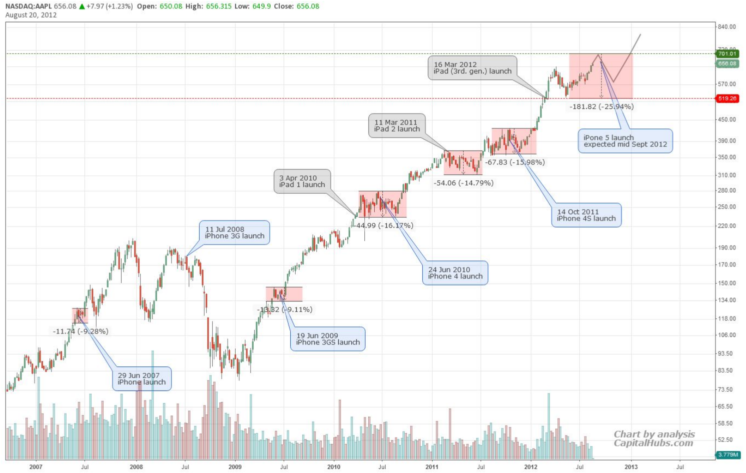
# Data Description




- Signal Description
  - 16 Channel EEG signals
  - Sampling rate: 400 Hz
  - Number of patients: 3
  - 1 hour signals are splitted into 6 segments
  - Original signal size: 73.9 GB
- How can we condense the information?

# What is Time Series?

- Definition
  - a sequence of data generated from discretized dynamic equations
- Characteristics
  - Correlation in time axis
- Applications
  - Sales
  - Biomedical Signals
  - Financial Data
  - Audio
  - ……



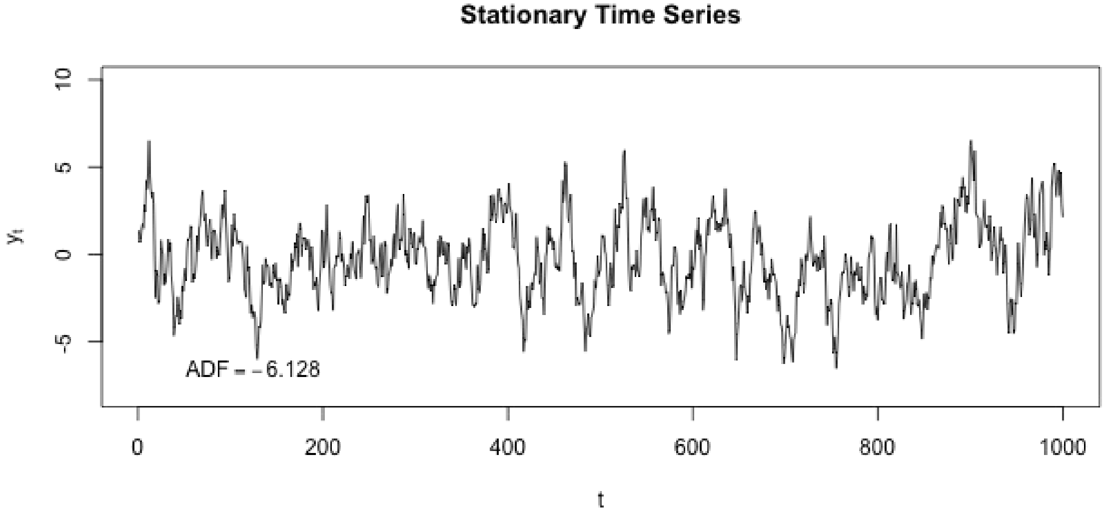
Source: TradingView.com

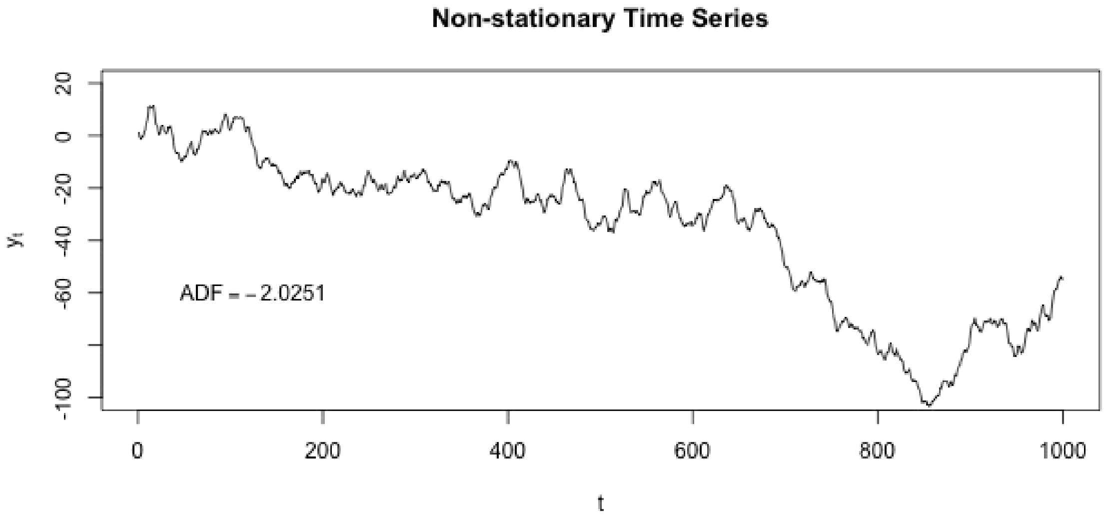
# Extract Information from Time Series

- Summary Statistics
- Frequency Domain
- Multiple Channels Features
  - Eigenvalue of Correlation Matrix (Time/Spectral)
- Model Based
  - AR Model

# Summary Statistics

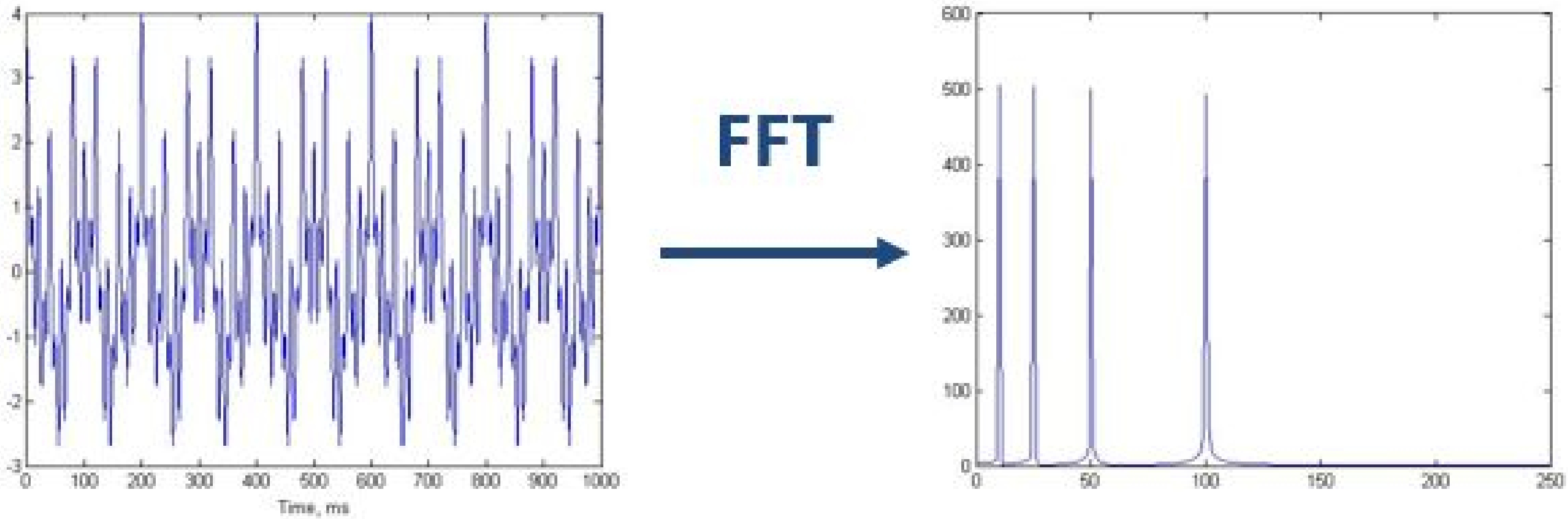
- Mean
- Variance
- Skewness
- Kurtosis



# Fourier Transform

- Transform time domain signal to frequency domain $\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx$

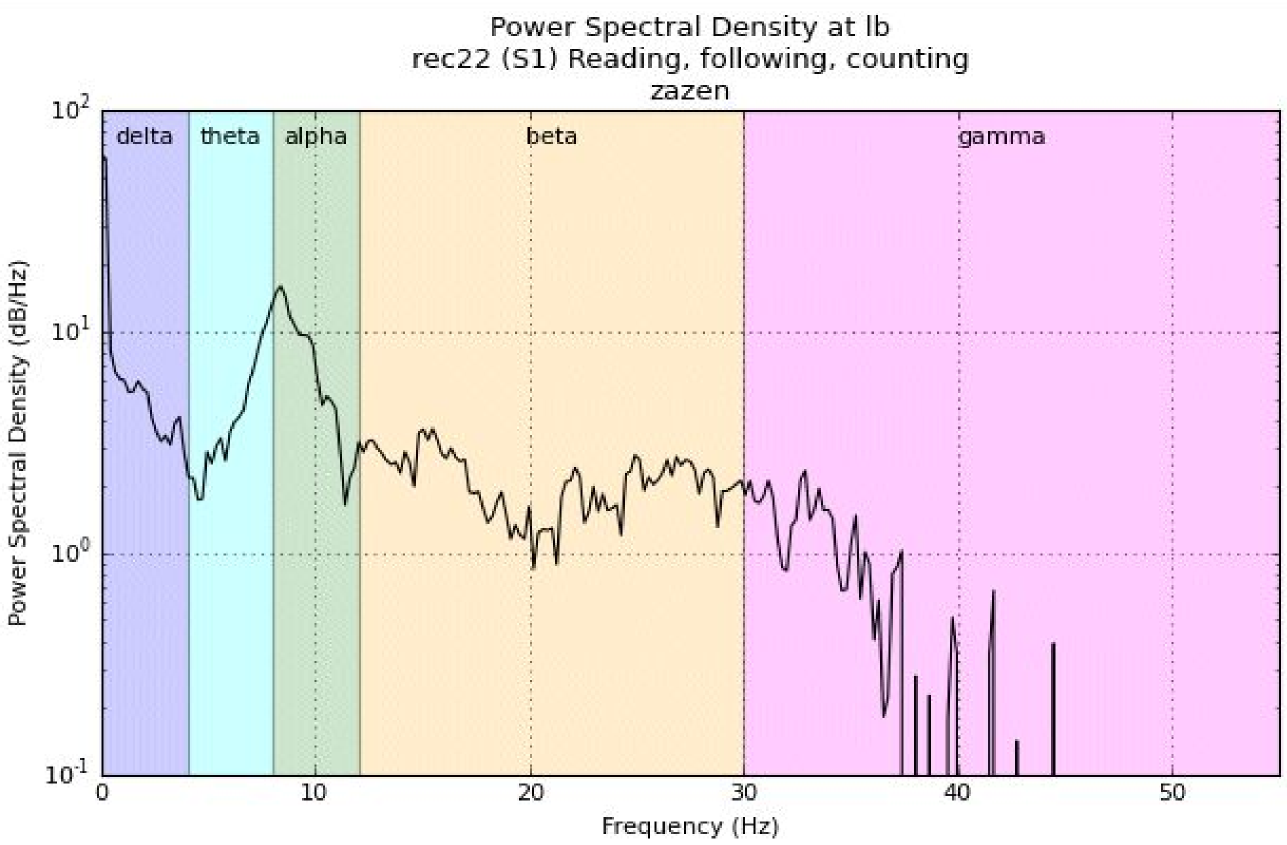



$$x(t) = \cos(2\pi * 10t) + \cos(2\pi * 25t) + \cos(2\pi * 50t) + \cos(2\pi * 100t)$$

**10, 25, 50, 100Hz**

# EEG Frequency Domain Analysis

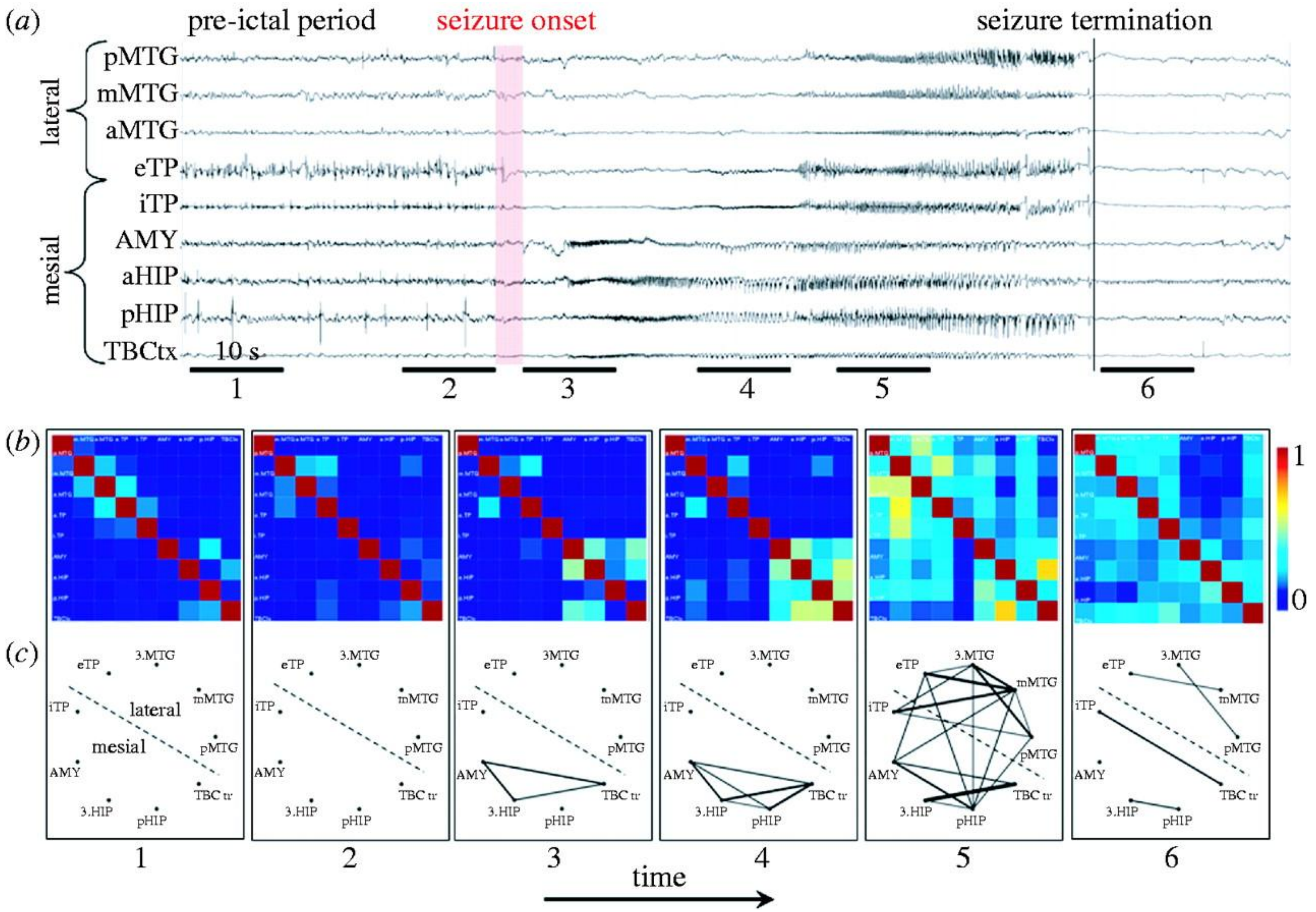
- Frequency band energy
  - predefined band
  - dyadic band
- Spectral edge frequency
  - The frequency below which x percent of the total power of a given signal are located




- delta: 0.1~4, theta: 1~4, alpha: 4~8, beta: 15~30, low gamma: 30~90, high gamma: 90~170

# Multiple Channels

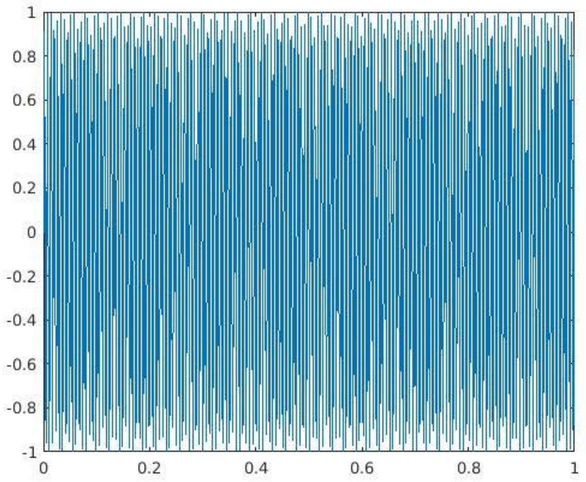
- Before seizure happens, some EEG channels will show join-excitation
- Use Correlation matrix in time/frequency domain and their eigenvalues



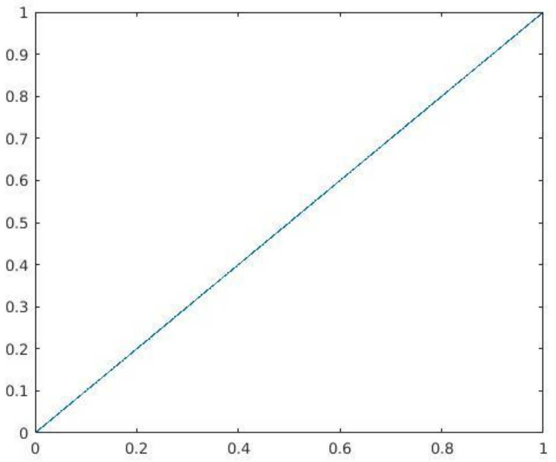
# Model Based

- Winner team uses AR coefficients as features
- For a set of observations: y(1), y(2)…….,y(N)

$$y_t = c + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \varepsilon_t$$



Coefficient: [- 1.081, 1]
Pole: 0.5405 + 0.8413i and 0.5405 - 0.8413i



Coefficient: [-2, 1]
Pole: 1 and 1

## Stationary signals

### Time domain

- Statistical-based
  - RMS
  - Variation
  - Skewness
  - Kurtosis
  - crest factor
- Model-based
  - AR model
  - HMM model
- Signal processing
  - TSA
  - Correlation
  - Convolution
  - Fractal analysis
  - Correlation dimension

### Frequency domain

- Spectral analysis
- Envelope analysis
- Cepstrum analysis
- Higher order spectrum

## Non-stationary signals

### Time-frequency

- Short-time Fourier Transform (STFT)
- Wigner-Ville distribution (WVD)
- Empirical mode decomposition (EMD)
- Basis pursuit
- Spectral kurtosis
- Cyclostationary analysis

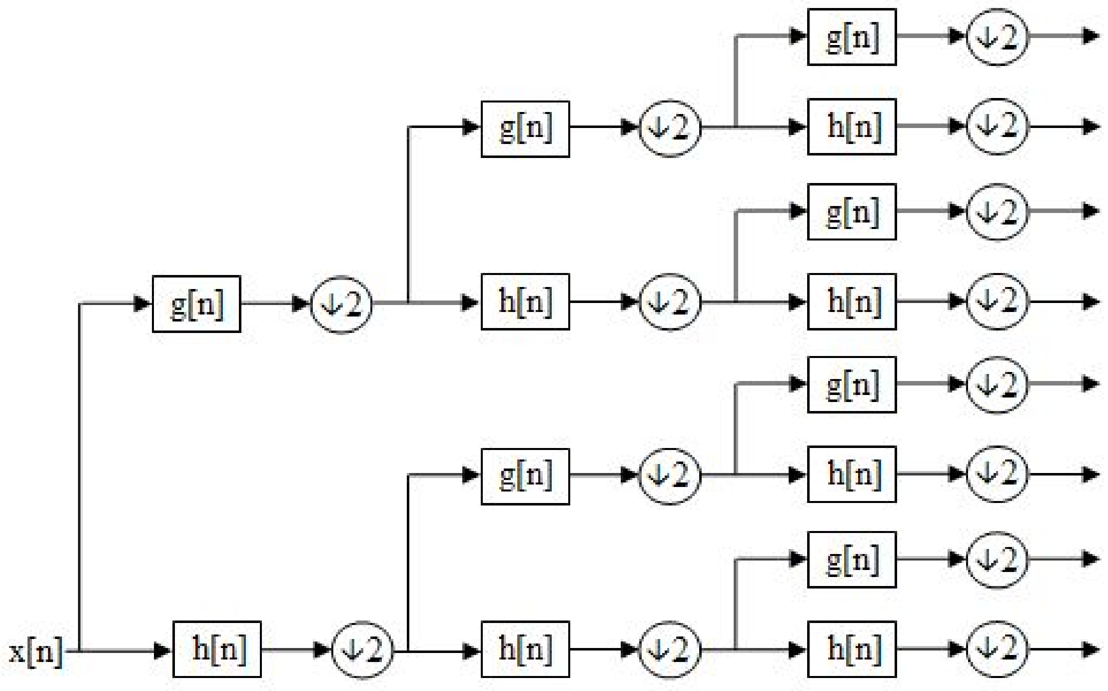
### Wavelets

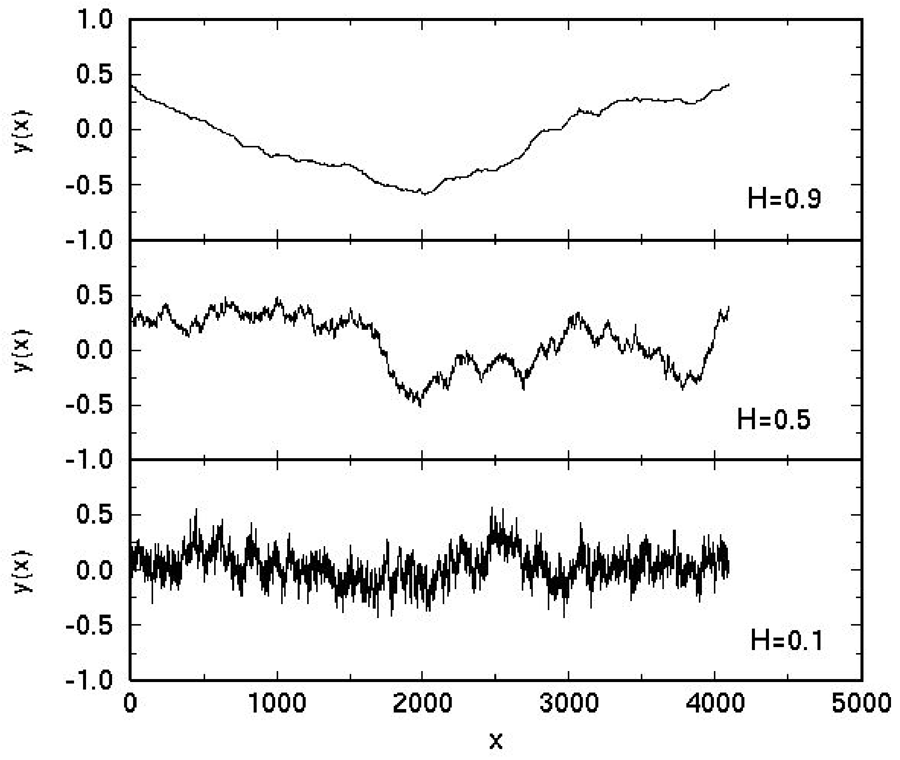
- Continuous wavelet transform (CWT)
- Discrete wavelet transform (DWT)
- Wavelet packet transform
- Morlet wavelet
- Hilbert-Huang transform

Yan, W. et al, 2008

# Feature used but not covered

- Wavelet packet coefficients
- Hurst exponent



# Where to find features?

- Google Scholar
- Previous Competitions
- Github
- Books
- ……

# After feature extraction

The extracted features are provided due to the time constraint. There are some more steps to do.

- Handle missing data: Fill nan using training data mean
- Identify zero signals: Remove all-zero signals

# From loading data to submission

- Datawarehouse: A class to handle data exchange
  - Read and process data
  - Select features
  - Generate submission files
- Model: A class to store machine learning models
  - Training models
  - Cross validation

# What happens to the private leaderboard

| # | △1w | Team Name ✱ in the money | Kernel | Team Members | Score | Entries | Last |
|---|---|---|---|---|---|---|---|
| 1 | ▲1 | ✱ Not-so-random-anymore | | | 0.80701 | 260 | 3mo |
| 2 | ▲35 | ✱ Areté Associates | | | 0.79898 | 56 | 3mo |
| 3 | ▲12 | ✱ GarethJones | | | 0.79652 | 74 | 3mo |
| 4 | ▲23 | QingnanTang | | | 0.79458 | 62 | 3mo |
| 5 | ▲11 | nullset | | | 0.79363 | 119 | 3mo |
| 6 | ▲14 | tralala boum boum pouêt pouêt | | | 0.79197 | 57 | 3mo |
| 7 | ▲7 | Medrr | | | 0.79183 | 89 | 3mo |
| 8 | ▲14 | michaln | | | 0.79074 | 48 | 3mo |
| 9 | ▼8 | DataSpring | | | 0.79053 | 55 | 3mo |
| 10 | ▼5 | fugusuki | | | 0.78773 | 82 | 3mo |

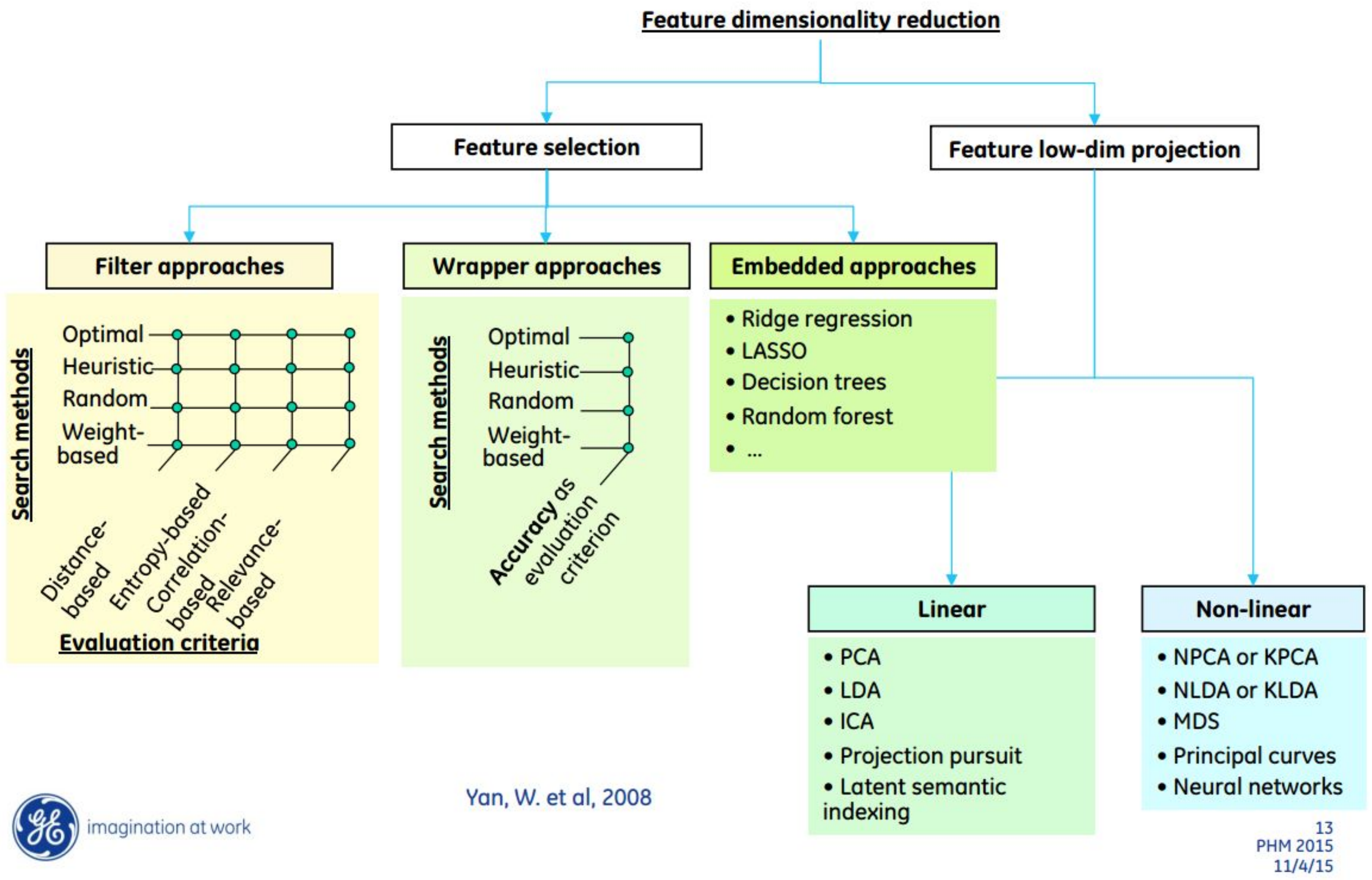
Net LB Gain for top 10: **104**

# Cross Validation Methods

- K-fold
- Leave p out
- Based on group
- Sequential

Which one is suitable for our data?

# Feature Selection



# Filter Approaches

## `sklearn.feature_selection`: Feature Selection

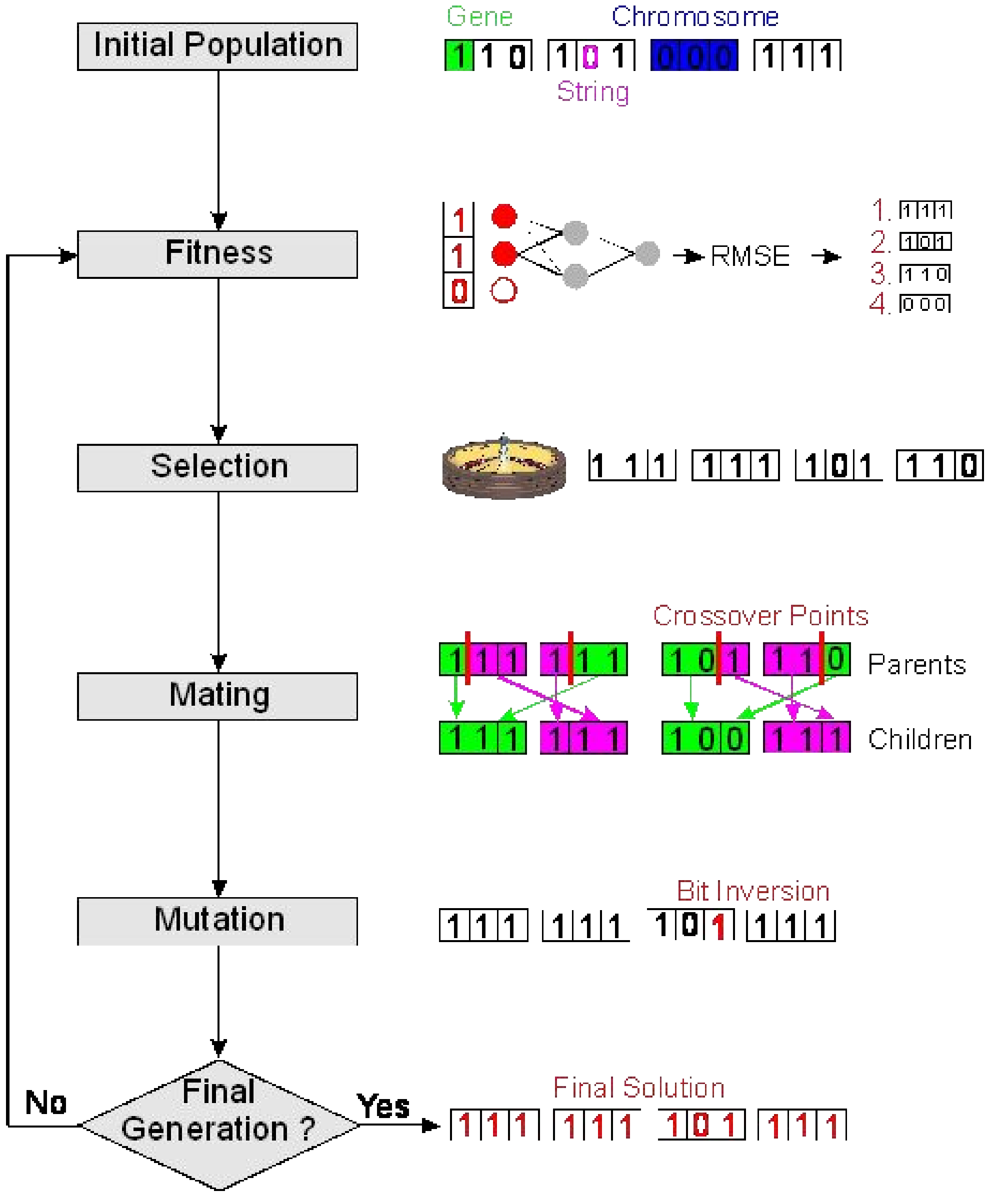
The `sklearn.feature_selection` module implements feature selection algorithms. It currently includes univariate filter selection methods and the recursive feature elimination algorithm.

**User guide:** See the Feature selection section for further details.

| | |
|---|---|
| `feature_selection.GenericUnivariateSelect` ([...]) | Univariate feature selector with configurable strategy. |
| `feature_selection.SelectPercentile` ([...]) | Select features according to a percentile of the highest scores. |
| `feature_selection.SelectKBest` ([score_func, k]) | Select features according to the k highest scores. |
| `feature_selection.SelectFpr` ([score_func, alpha]) | Filter: Select the pvalues below alpha based on a FPR test. |
| `feature_selection.SelectFdr` ([score_func, alpha]) | Filter: Select the p-values for an estimated false discovery rate |
| `feature_selection.SelectFromModel` (estimator) | Meta-transformer for selecting features based on importance weights. |
| `feature_selection.SelectFwe` ([score_func, alpha]) | Filter: Select the p-values corresponding to Family-wise error rate |
| `feature_selection.RFE` (estimator[, ...]) | Feature ranking with recursive feature elimination. |
| `feature_selection.RFECV` (estimator[, step, ...]) | Feature ranking with recursive feature elimination and cross-validated selection of the best number of features. |
| `feature_selection.VarianceThreshold` ([threshold]) | Feature selector that removes all low-variance features. |
| `feature_selection.chi2` (X, y) | Compute chi-squared stats between each non-negative feature and class. |
| `feature_selection.f_classif` (X, y) | Compute the ANOVA F-value for the provided sample. |
| `feature_selection.f_regression` (X, y[, center]) | Univariate linear regression tests. |
| `feature_selection.mutual_info_classif` (X, y) | Estimate mutual information for a discrete target variable. |
| `feature_selection.mutual_info_regression` (X, y) | Estimate mutual information for a continuous target variable. |

# Wrapper Approaches - Genetic Algorithm

- Use binary vector to represent feature selection
- Randomized selection algorithm
- Computational Intensive



# Embedded Approaches

Let's try L1 regularized logistic regression

# Comparison

| Model | Local | Public | Private |
|---|---|---|---|
| L1 Regularized LR | 0.79080 | 0.63398 | 0.65289 |
| Random Forest | 0.82348 | 0.75597 | 0.72346 (rank 66, bronze) |
| Random Forest with L1 | 0.82277 | 0.75811 | 0.72965 (rank 66, bronze) |

# Parameter Tuning

- Graduate student descent
- Grid search
- Random search (Genetic algorithm can also be used here)
- Bayesian optimization