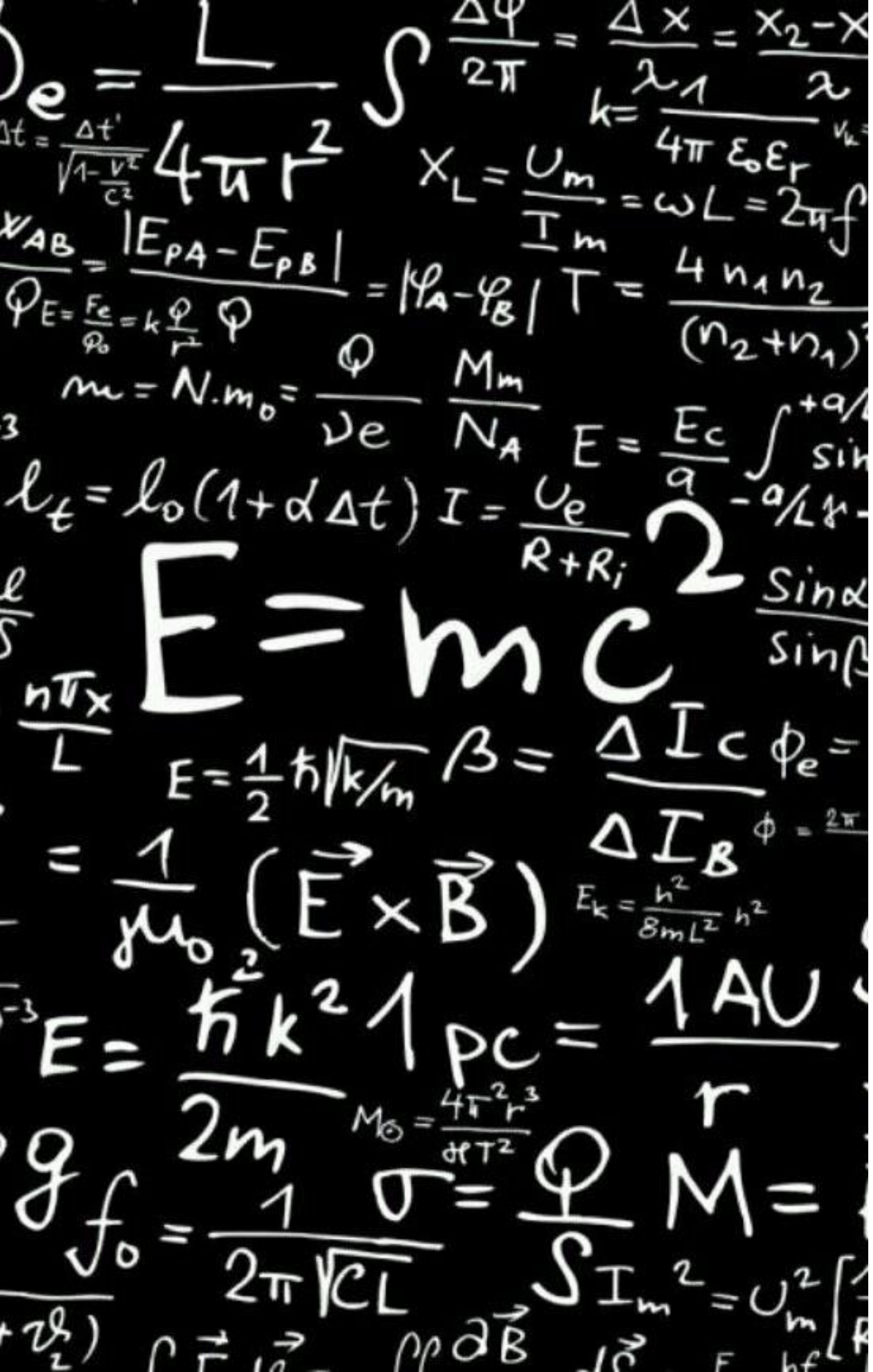


DATA PREPROCESSING And FEATURE ENGINEERING

Why do we care?

- ➤ **Data Preprocessing**
  - Real world data almost always comes as a mess
  - Some models have specific assumptions about inputs

- ➤ **Feature Engineering**
  - Features in inputs sometimes are not enough
  - Insights from human knowledge or domain knowledge can improve accuracy

Data Preprocessing

- **Data Cleaning**: Real-world data is dirty — incomplete and inconsistent. We will need to perform cleaning operations such as filling in missing values, removing some outliers.
- **Data Integration**: We might need to pull data from several sources and make sure they are consistent.
- **Data Transformation**: We might need to transform data to fit machine learning model assumption such as scaling and standardizing.
- **Feature Selection**: Many features may be useless. Filtering irrelevant features out may improve the result.
- **Feature Extraction**: We need to transform data to help our machine learning model to learn from it, such as image processing.

Data Preprocessing

➤ **Data Transformation**: We might need to transform data to fit machine learning model assumption such as scaling and standardizing.

Problems with the dataset:

- Different scale
- Different distribution
- Skewed data
- Categorical data

Data Preprocessing

➤ **Data Cleaning**: Real-world data is dirty — incomplete and inconsistent. We will need to perform cleaning operations such as filling in missing values, removing some outliers.

Problems with the dataset:

- Missing Values
- Outliers

Data Preprocessing

➤ **Data Integration**: We might need to pull data from several sources and make sure they are consistent.

Problems with the dataset:

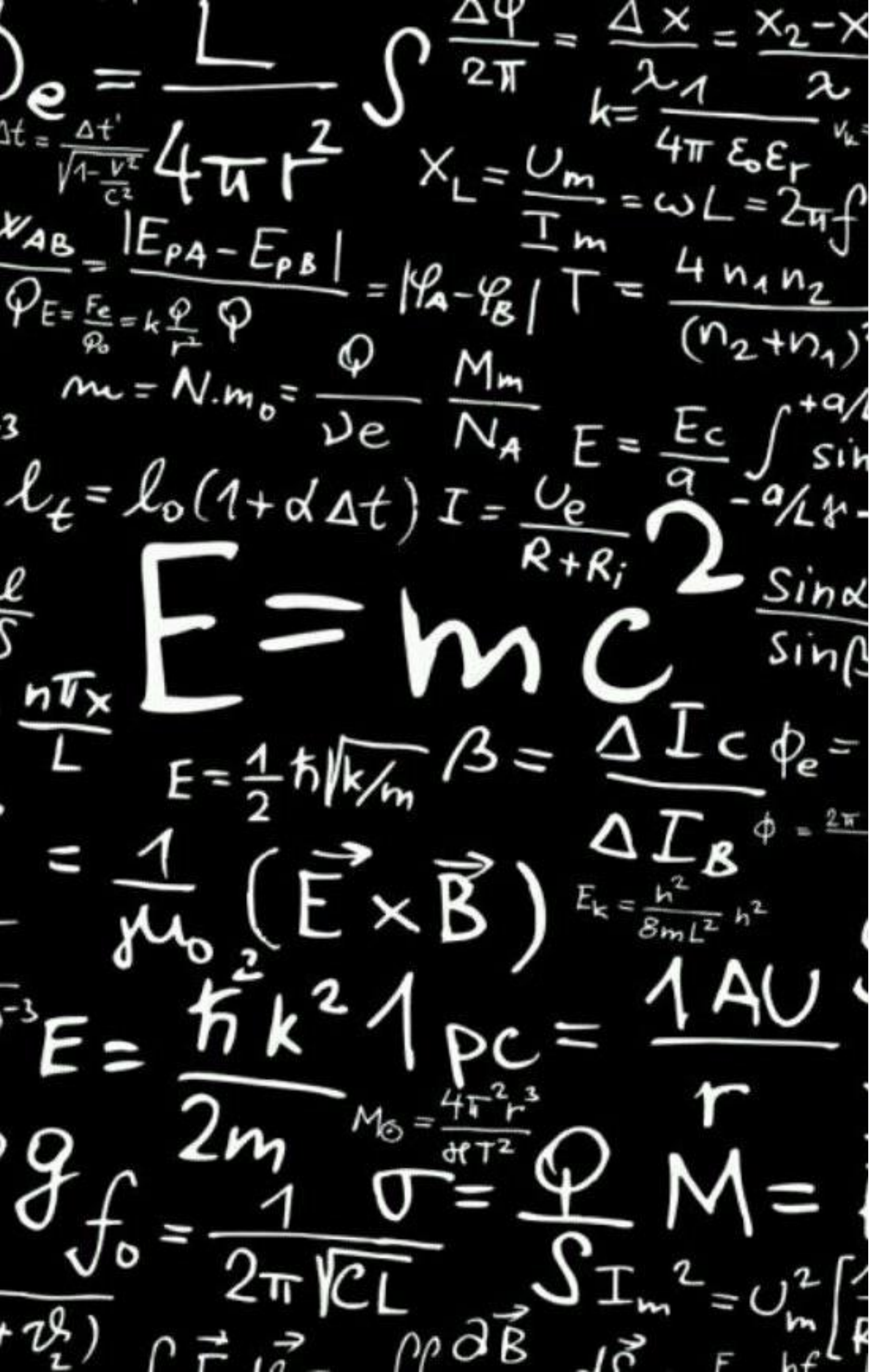
- Needed information are distributed across many tables

Data Preprocessing

- ➤ **Feature Selection**: Many features may be useless. Filtering irrelevant features out may improve the result.
- ➤ **Feature Extraction**: We need to transform data to help our machine learning model to learn from it, such as image processing.

Problems with the dataset:

- Irrelevant features
- Curse of dimensionality

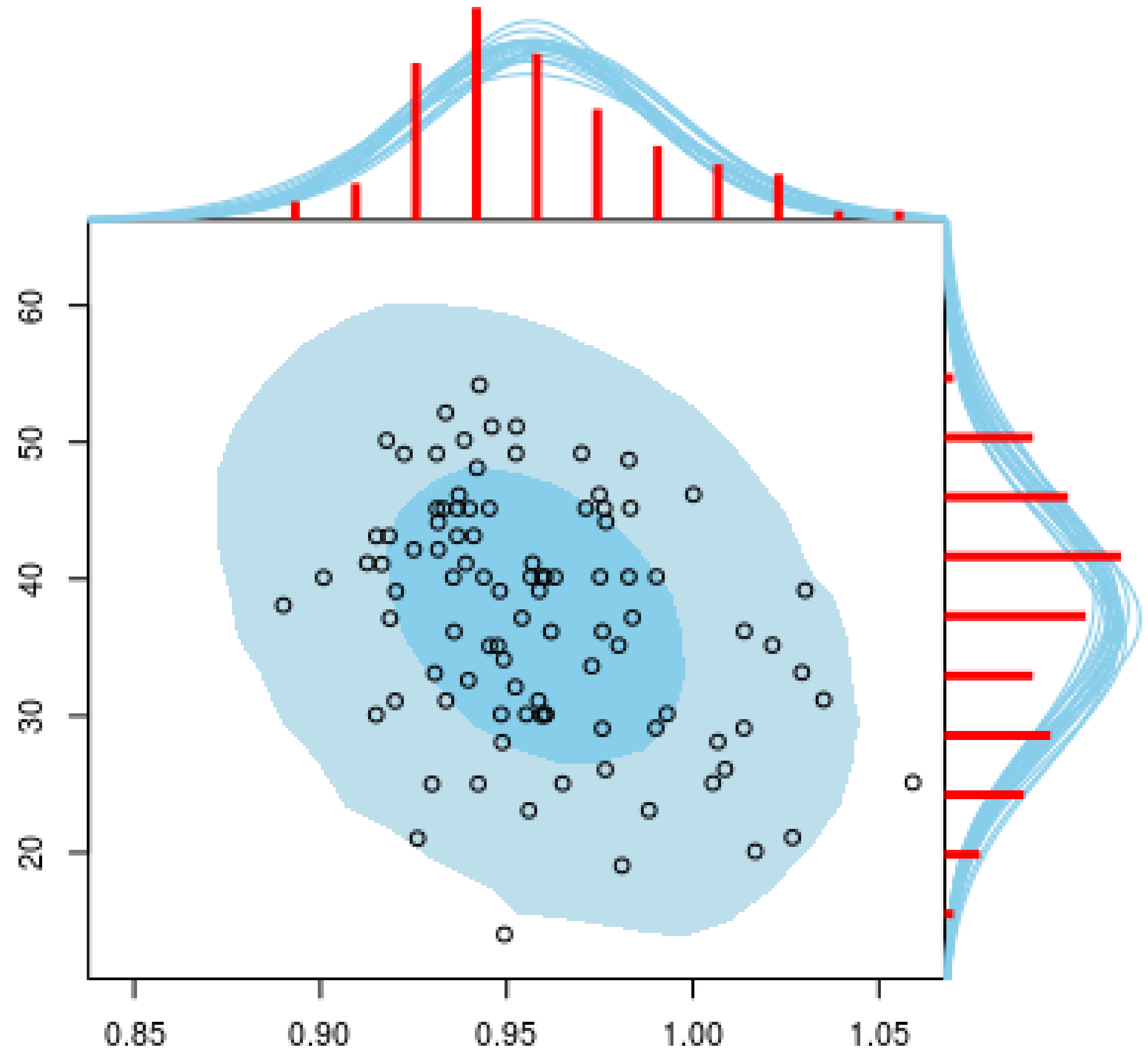


DATA TRANSFORMATION

Model Assumption

It’s important to be aware of model assumptions, when you use them.

- For example, regression has the following assumptions:
  - X and Y variables are numeric
  - X and Y variables have normal distributions
  - All variables must have the same variance



Feature Scaling

- Imagine if x1 and x2 are not similar in scale
- For example
  - x1 = number of bedrooms (0-20)
  - x2 = area of the house (24-3000 sqm)
- This means the scale of your theta will also be different.

Why Scaling

- Some algorithms like regression models and neural networks are very sensitive to scales.
- For example, if you want to compare the impact of number of bathrooms (scale 1-4) to the impact of areas (scale 24-250 sqm) on house prices, how do you make sure you are comparing apples or oranges? You need to rescale!

Feature Scaling

normalization with max, min, mean

$$x_1 := \frac{x_1}{max(x_1)}$$

$$x_1 := \frac{x_1 - mean(x_1)}{max(x_1)}$$

$$x_1 := \frac{x_1 - mean(x_1)}{max(x_1) - min(x_1)}$$

standardization
a.k.a. z-score

$$x_1 := \frac{x_1 - mean(x_1)}{std(x_1)}$$

What's the range of these rescaled variables?

One-hot encoding

- When we have categorical data and our model requires numerical data, we need to transform text to numbers to feed into the model.
- A one-hot encoding operation transform categorical data into columns of 0 and 1.

Sample	Color
1	Red
2	Green
3	Blue
…	…

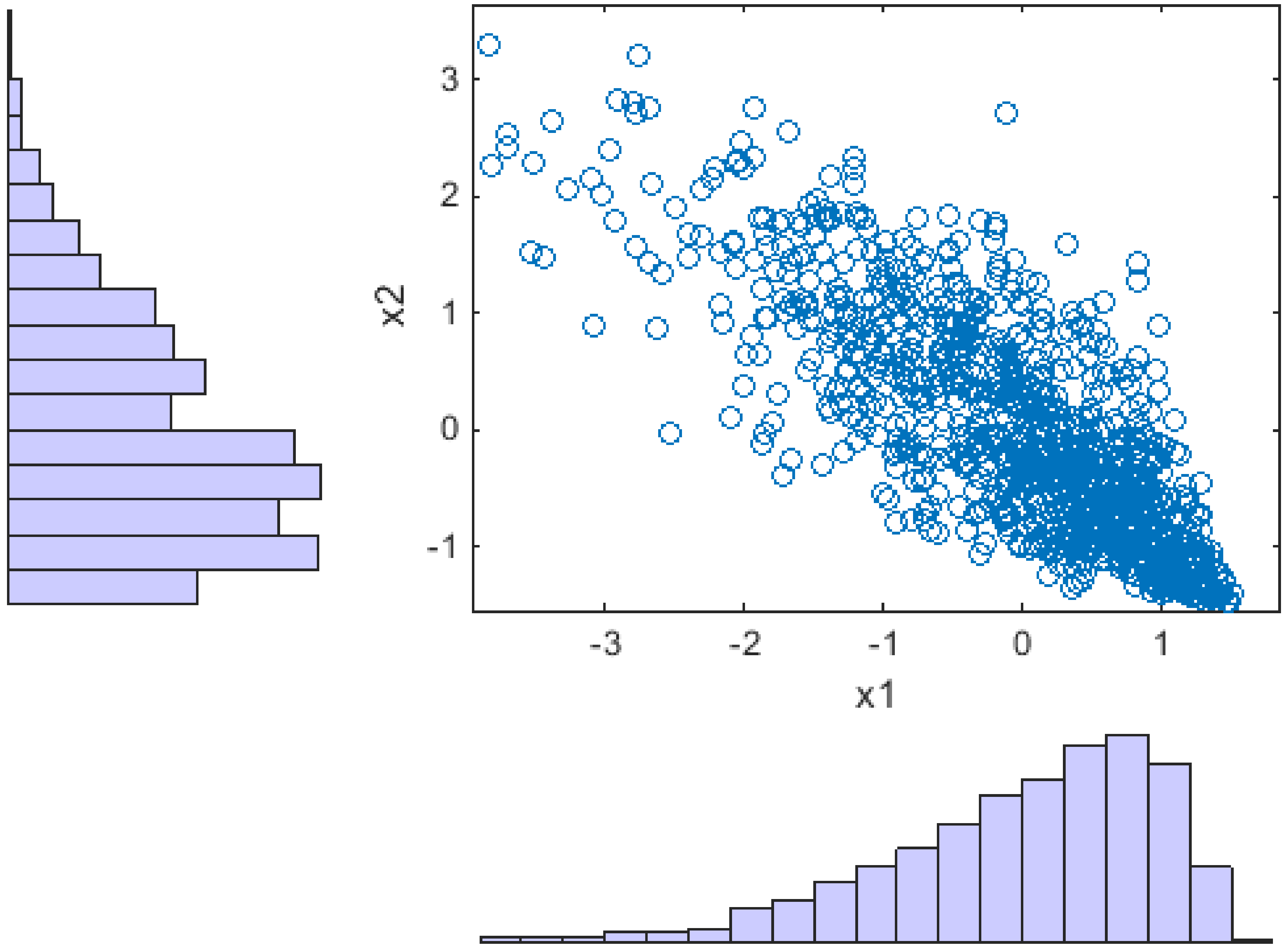
➡

Sample	Is_Red	Is_Green	Is_Blue
1	1	0	0
2	0	1	0
3	0	0	1
…	…	…	…

Let’s try this

Sample	Grade
1	A
2	B
3	A
4	B+
5	F
6	D+
7	B
8	C
9	D+

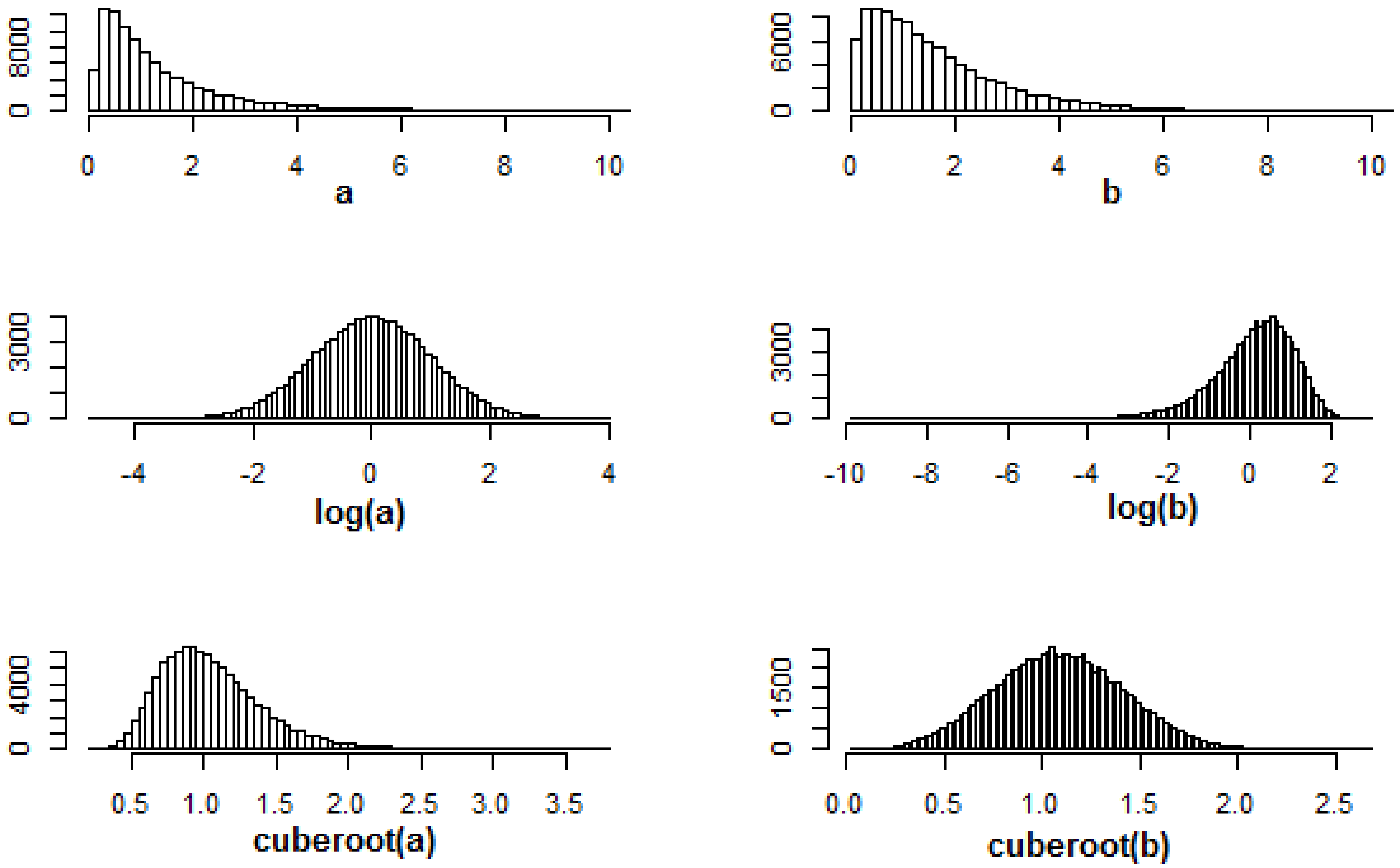
Skewed Data



- Some algorithms like linear regression model assume normally distributed inputs.

- Many features do not behave that way!

Correcting Skewed data



➤ Applying non-linear function can fix skewed data

- Logarithm function
- Cube root
- Box-Cox

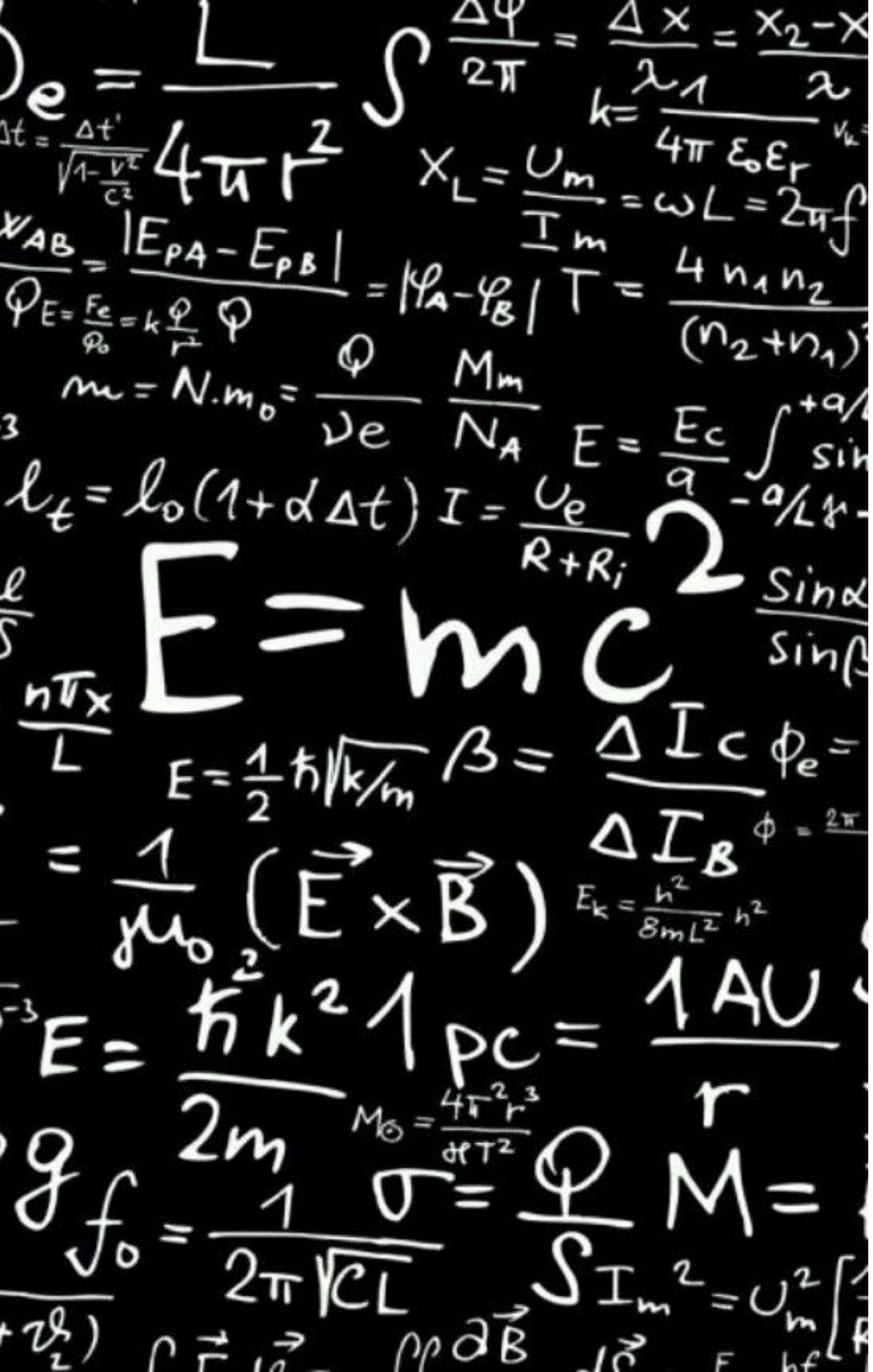
Let’s Try this

- Given the data:

1, 8, 8, 8, 27

What does the distribution look like?

How do we fix it?



DATA CLEANING

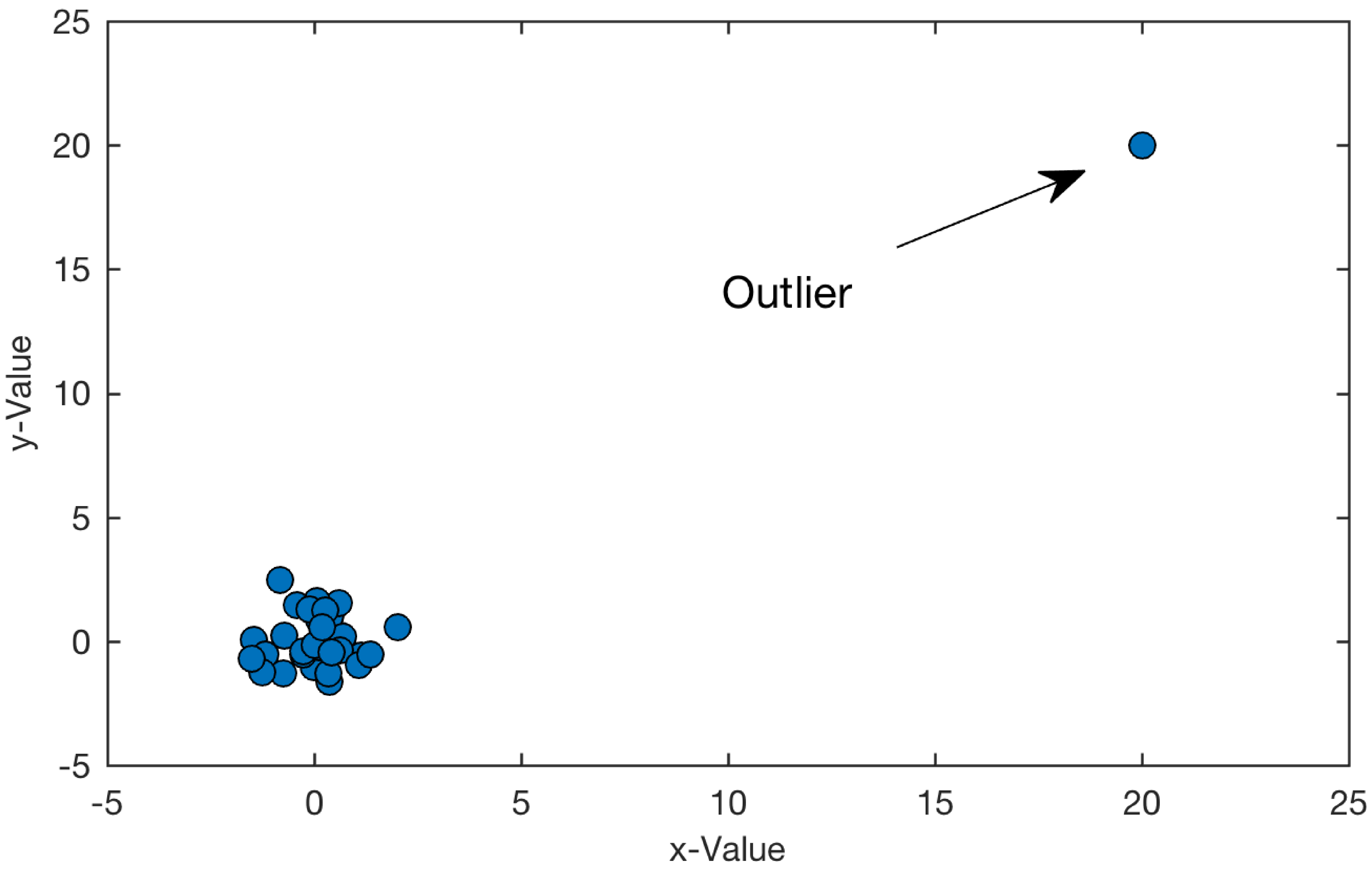
Missing Data

- If data are assumed to be missing at random, we may simply ignore the data
  - You went to all houses and randomly for some houses, you took time to do detailed house measurement
  - In this case, you simply remove the missing data from the analysis (cut the whole row or column)
  - Be aware that missing data might not be as random as you think

Missing Data

- Mean or mode substitution
  - Missing income? Fill it with average income?
  - Don’t know if the patient is left-handed or right-handed, assume right-handed, because it’s more common.

Outliers



- Outliers are those data points that are way different from the rest.
- Outliers can be misleading.
- For example, in this picture, without outliers you see no correlations in the data, but with outliers, correlations start to emerge, but those correlations are unreal!

What Cause Outliers?

- Measurement errors (mistyped data)
- Sensor errors
- Freaky circumstances (events that are unlikely, but happen, for example, a gigantic product sales due to liquidation).

Removing Outliers

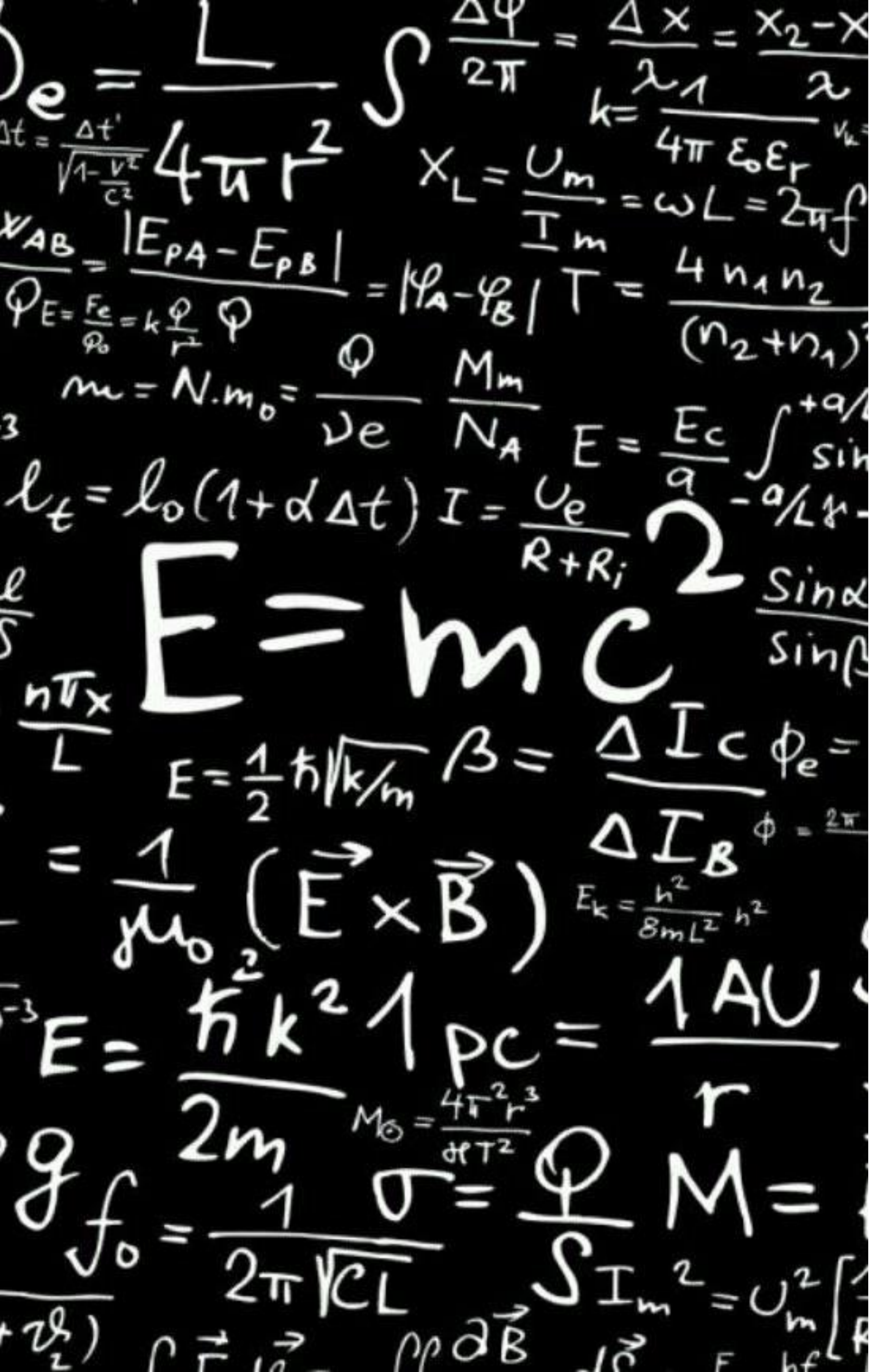
➤ Make a scatterplot or histogram plot of your data and look for extreme values.

➤ Assume that data has a Gaussian distribution and look for values more than 2-3 standard deviations from the mean.

➤ Filter out outliers candidate from training dataset and see if model’s performance improve.

Outlier Removal Algorithm

- Train model with all the data
- Look at data points with largest residual errors (say 10% of the data)
- Remove those data points from the dataset
- Retrain the model, do it over and over.



DIMENSIONALITY REDUCTION

Curse of Dimensionality

- This is one of the most important problems in machine learning.
  - Imagine you have a large amount of features, each can have infinite number of values.
  - You will need an enormous amount of training data is required to ensure that there are several samples with each combination of values.
  - If you have limited samples, which do not cover the whole space, your model loses predictive power.

Curse of Dimensionality

- This is one of the most important problems in machine learning.
    - Take linear regression for example
    - The more features you have, the more parameters you need to fit the model
    - If you have 2 features, your solution space has 2 dimensions (small possible values)
    - If you have 1000 features, your solution space has 1000 dimensions (huge amount of possible values.
    - You algorithm can take a lot of time to find solution.

Dimensionality Reduction

Dimensionality Reduction: take very high-dimensional features and transform them into lower-dimension features without losing the quality of the data.

This is usually done by optimizing a cost function that quantifies goodness of components.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{100} \end{bmatrix} \longrightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Methods:

Principle component analysis, singular value decomposition, independent component analysis

Dimensionality Reduction

Dimensionality Reduction

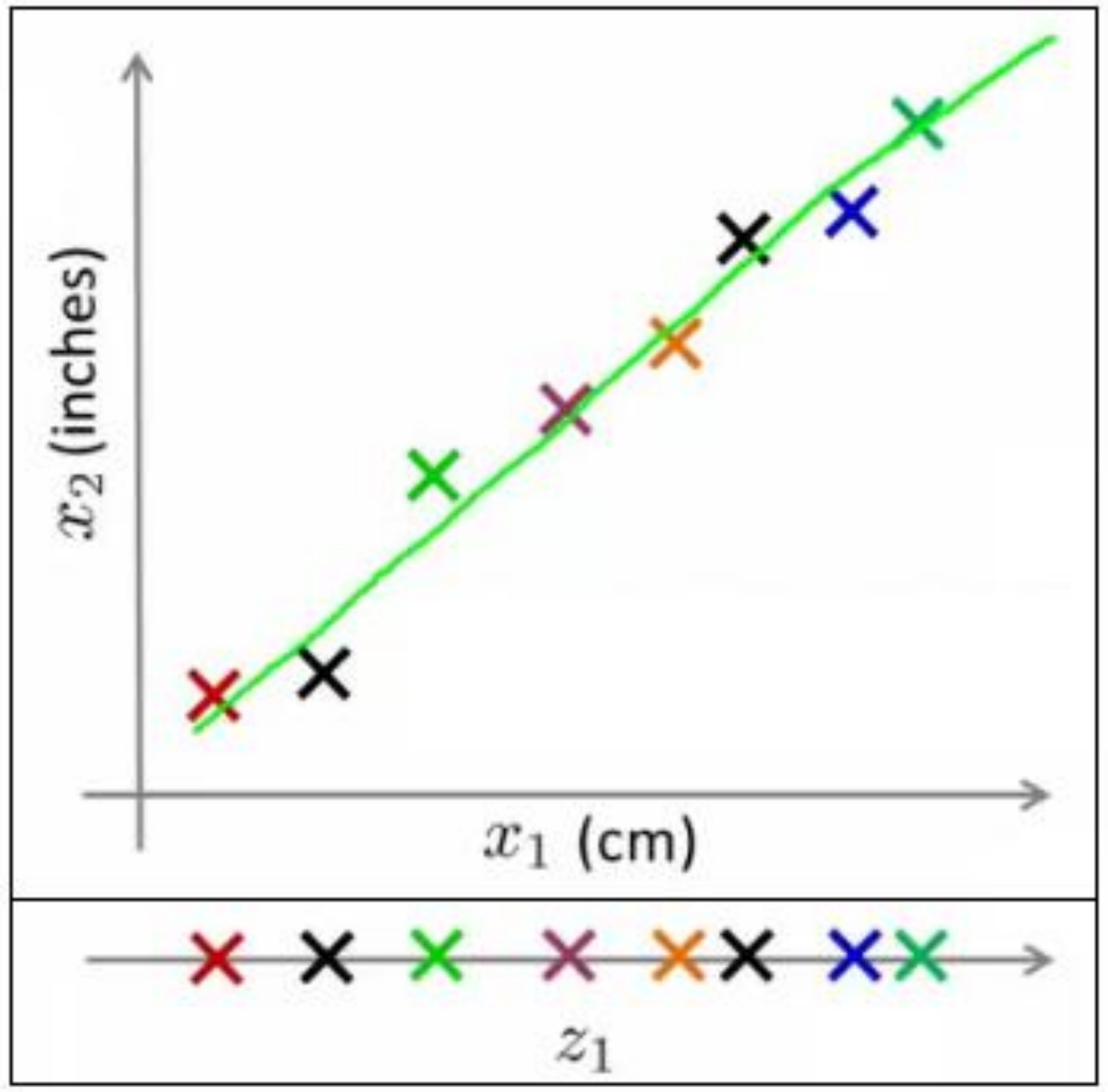
A process which takes very high-dimensional features and transform them into lower-dimension features without losing the quality of the data

Generally, this can be divided into

- Feature Extraction: Create few new features from old ones
- Feature Selection: Choose few important features

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_{100} \end{bmatrix} \longrightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Feature Extraction



PCA

- Starting with high-dimensional data
- Find new axes where data points can be projected on
- Get data on a new lower dimension form

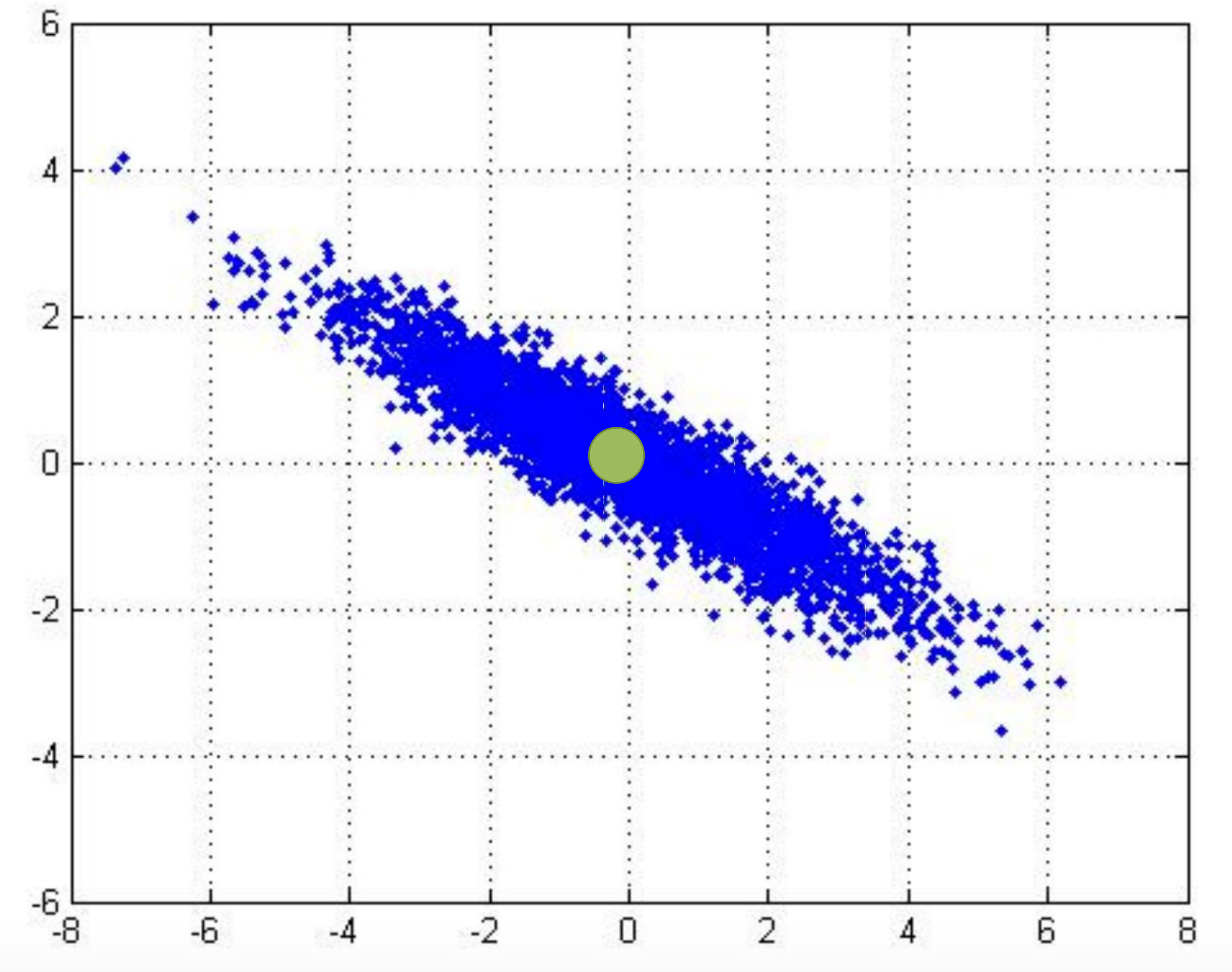
More Realistic Example

We collected restaurant ratings from 1M people rating 1000 restaurants (this dataset has 1000x1000 dimensions). We don’t want to build model in 1000 dimensions (too computationally expensive). We will need PCA for this one.

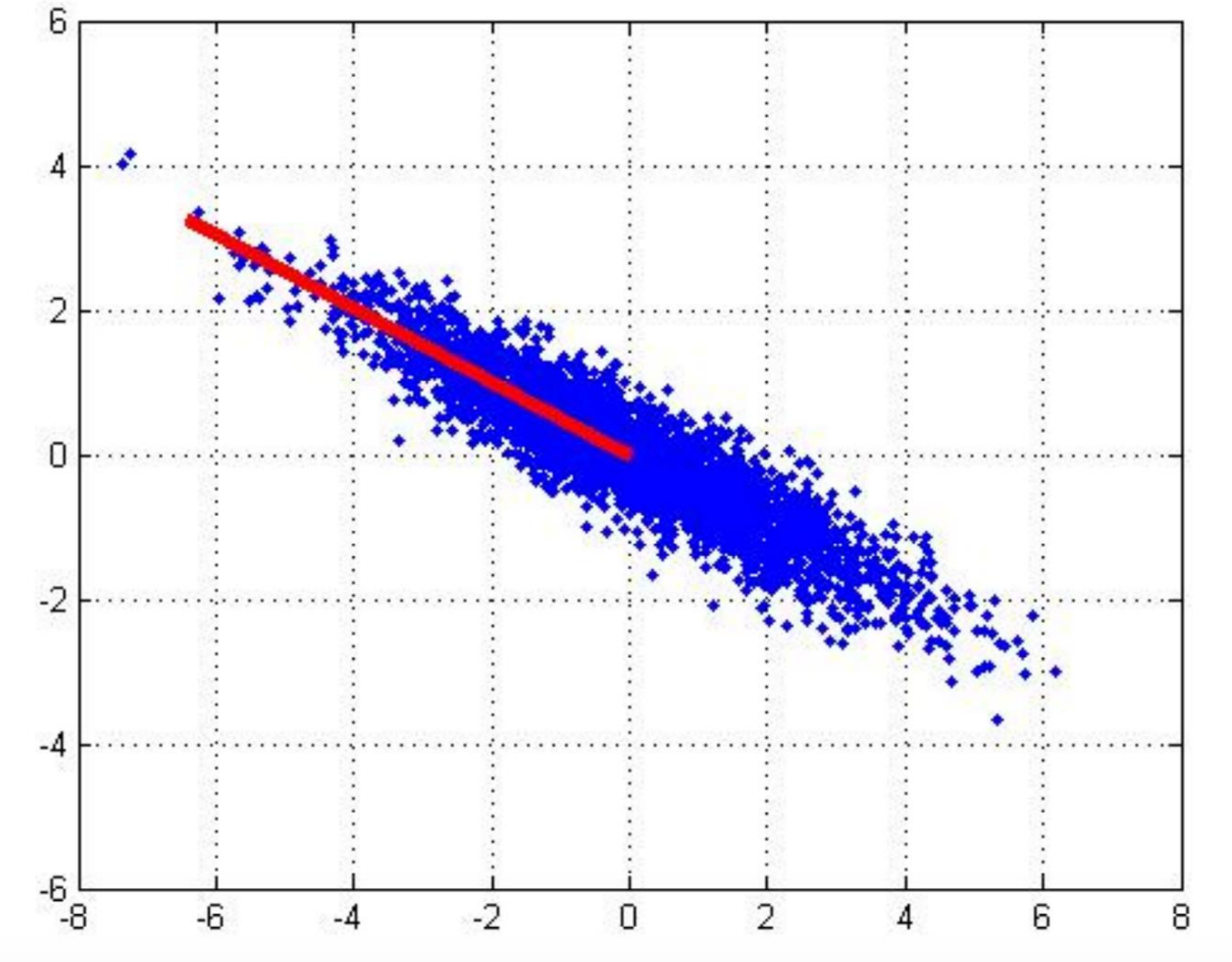
Why Principal Component Analysis

- ➤ Visualization
- ➤ More efficient use of resources
- ➤ Noise and outlier removal
- ➤ Faster processing by machine learning algorithm

Principle Component Analysis Algorithm



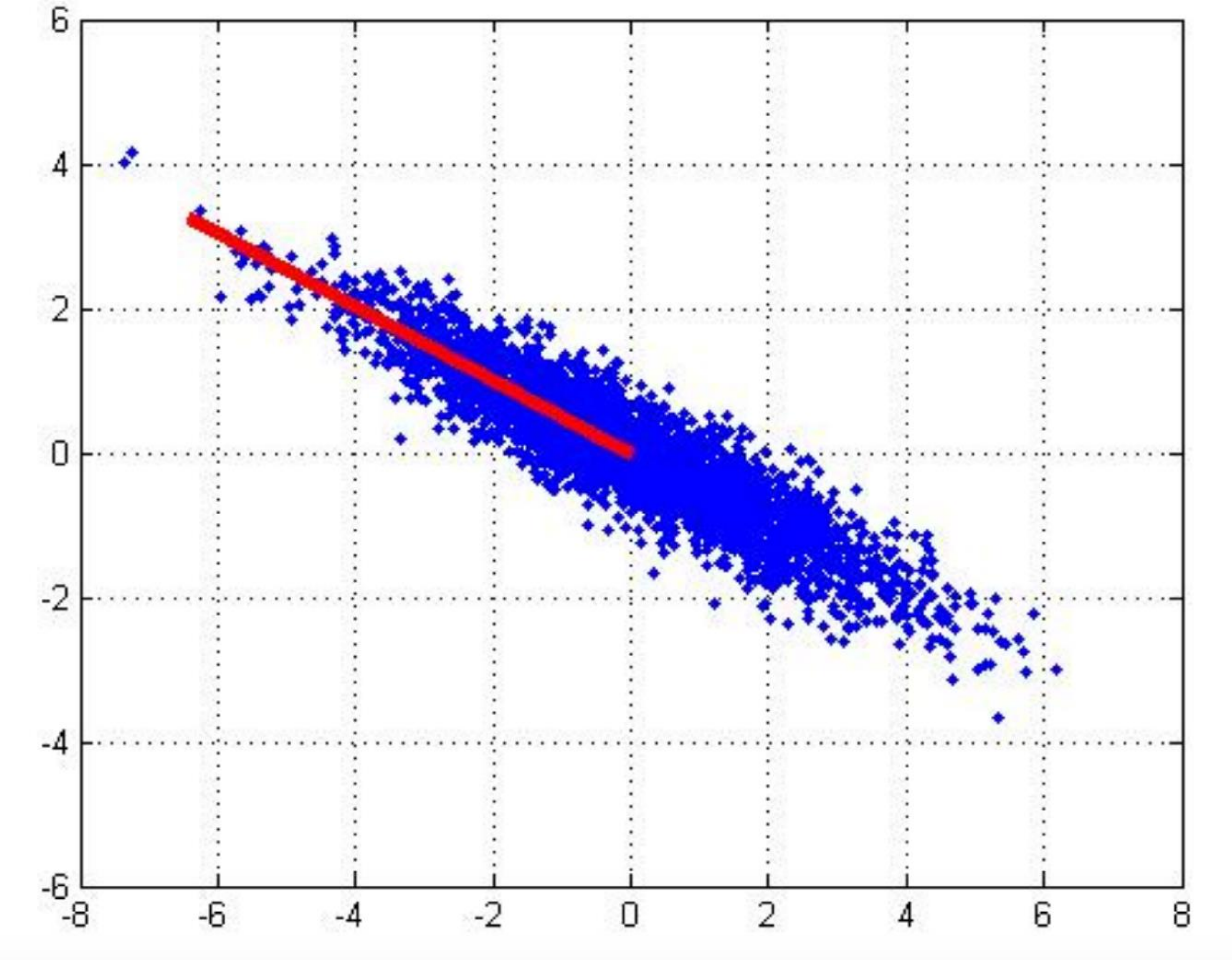
Start at the center of the data



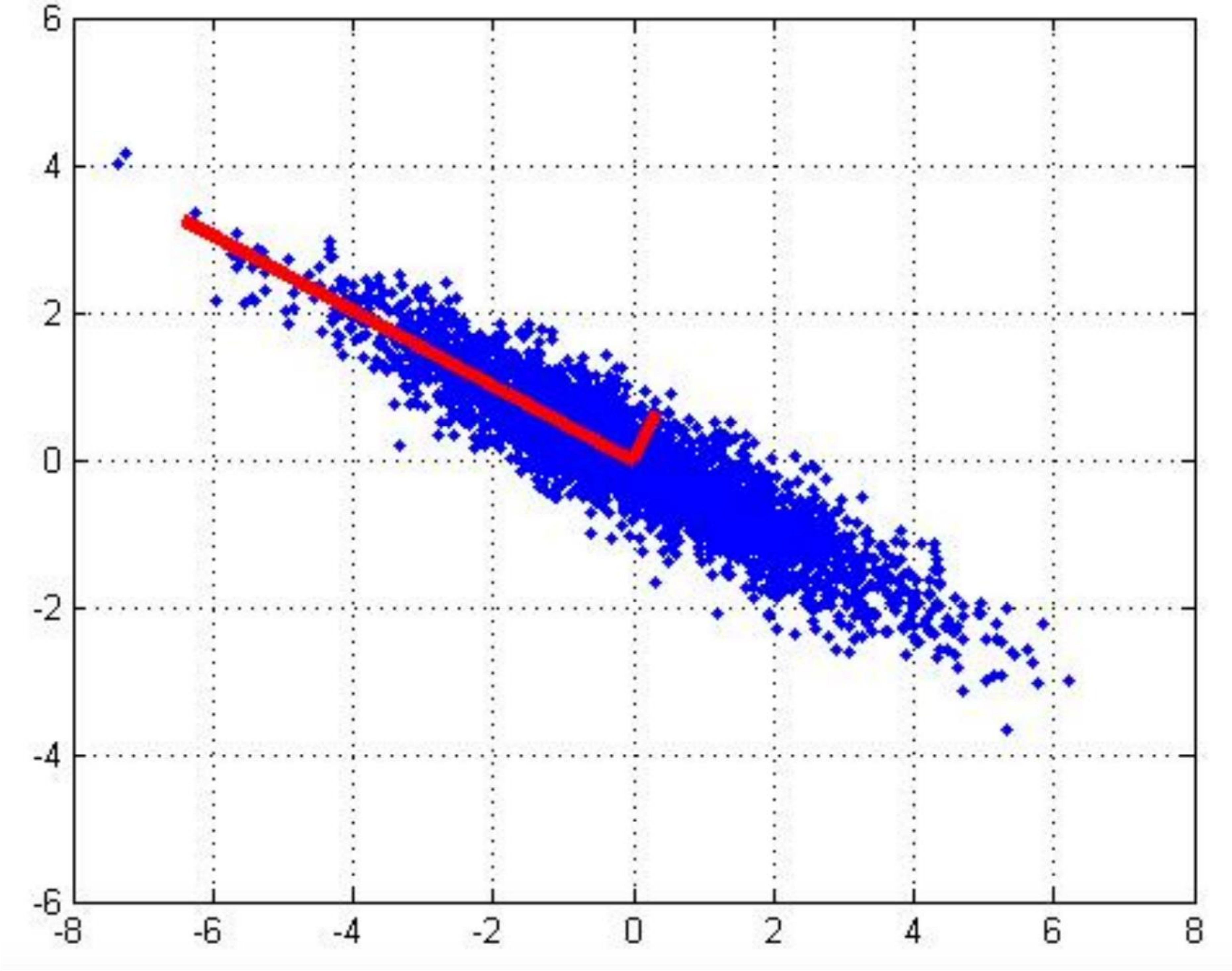
Find the dimension which maximizes variance.
That's the first principal component!

Slide of Matt Gormley

Principle Component Analysis Algorithm



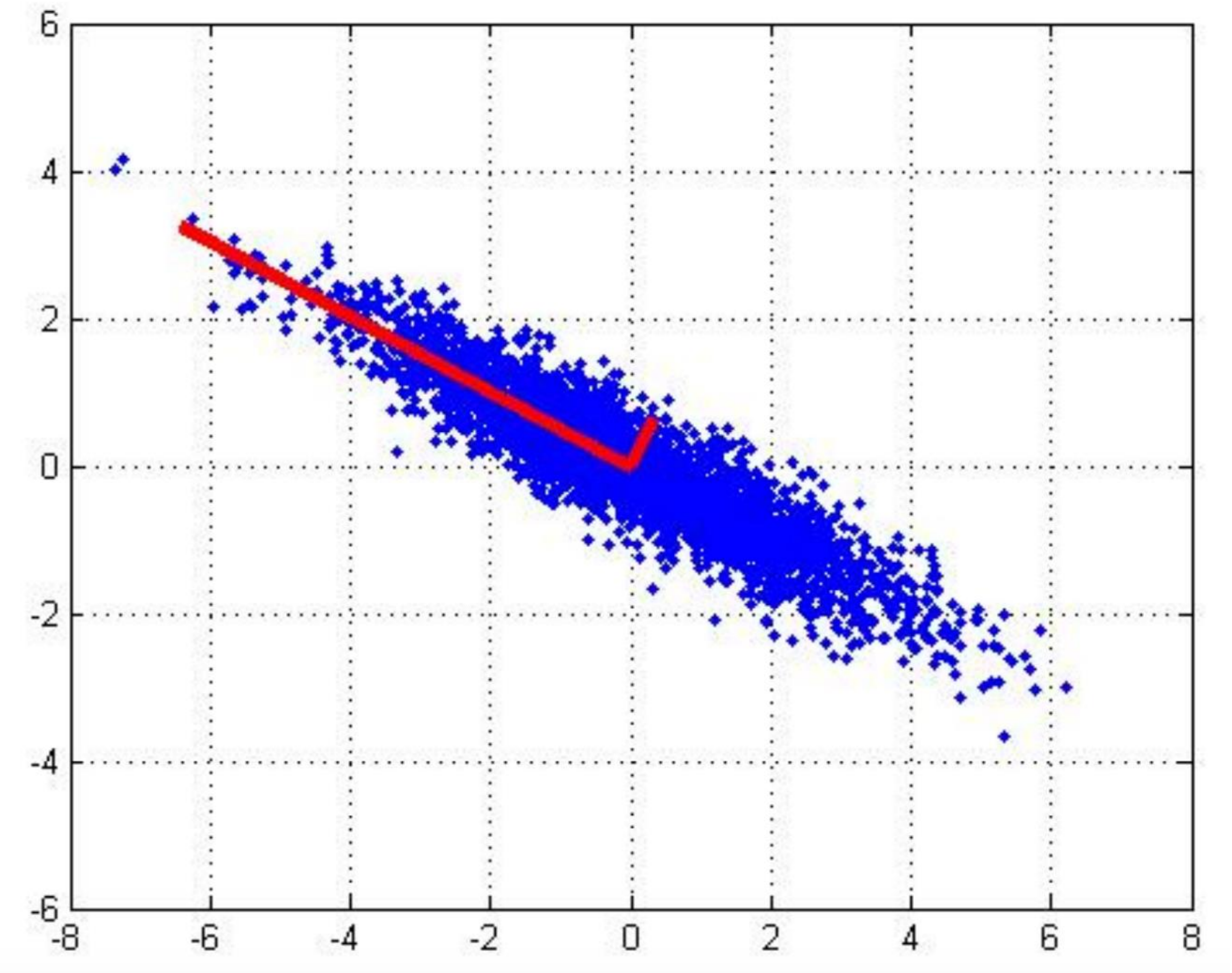
Find the dimension which maximizes variance.
That’s the first principal component!



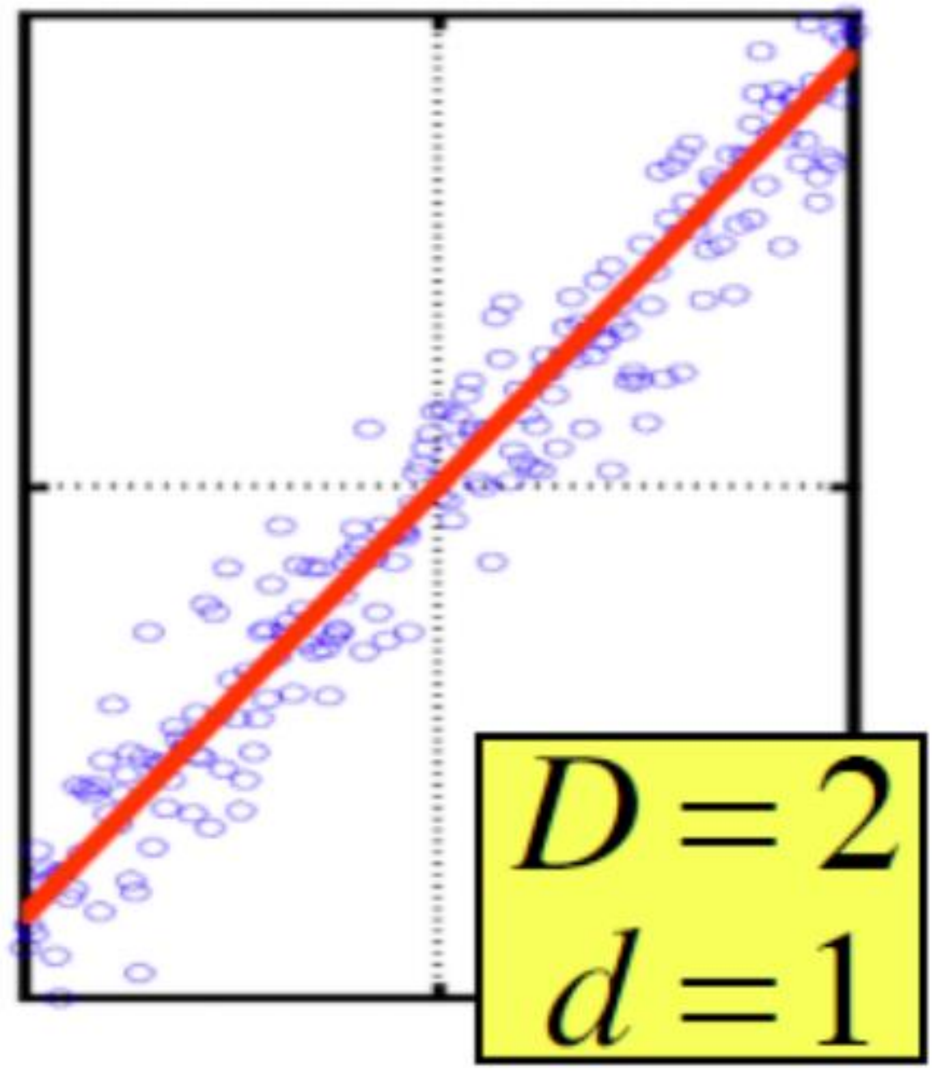
Find another dimension that is perpendicular to the first dimension, and maximize variance.
That’s the second principal component!

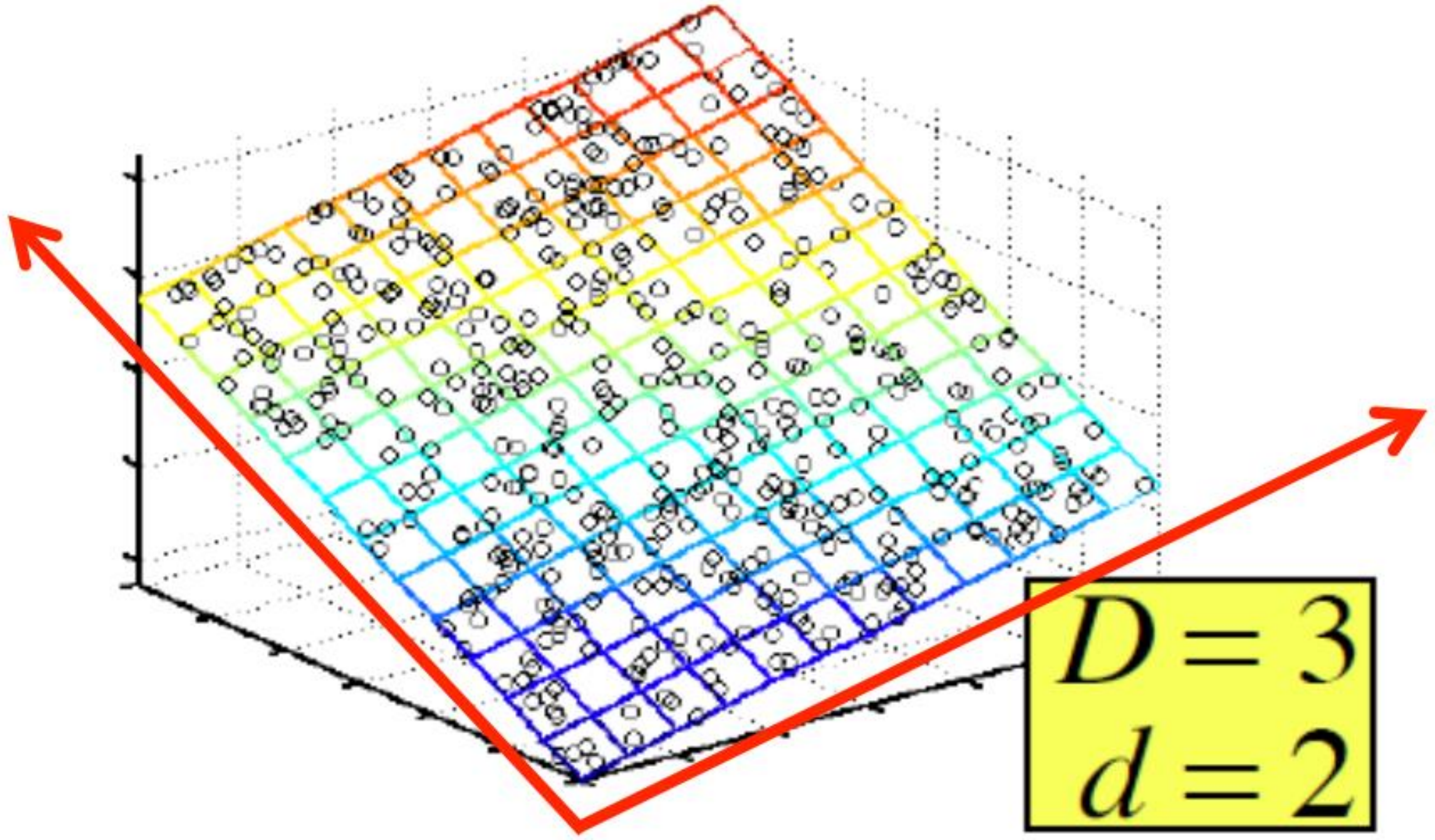
Slide of Matt Gormley

Principle Component Analysis Algorithm



Find another dimension that is perpendicular to the first dimension, and maximize variance. That’s the second principal component!





For N dimensions, continue until you reach the N component.

Slide of Matt Gormley

Big Data

- Most problems you will face in the real world is gigantic.
  - Millions of rows
  - Hundreds or thousands of features
  - Your algorithm will take forever to run
- What can we do about it?
  - We might be able to look through all the features and manually select them.
  - But that would waste so much time and resources
  - So maybe do automated feature selection?

Feature Selection

A process of selecting relevant, most important features to be used in machine learning model construction

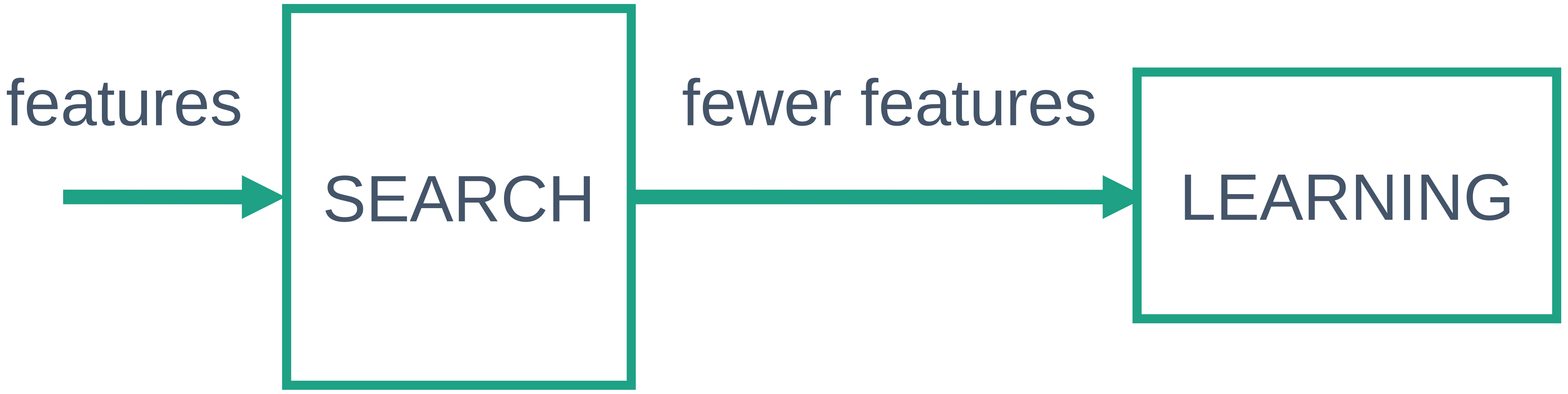
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_{100} \end{bmatrix} \longrightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Important Features

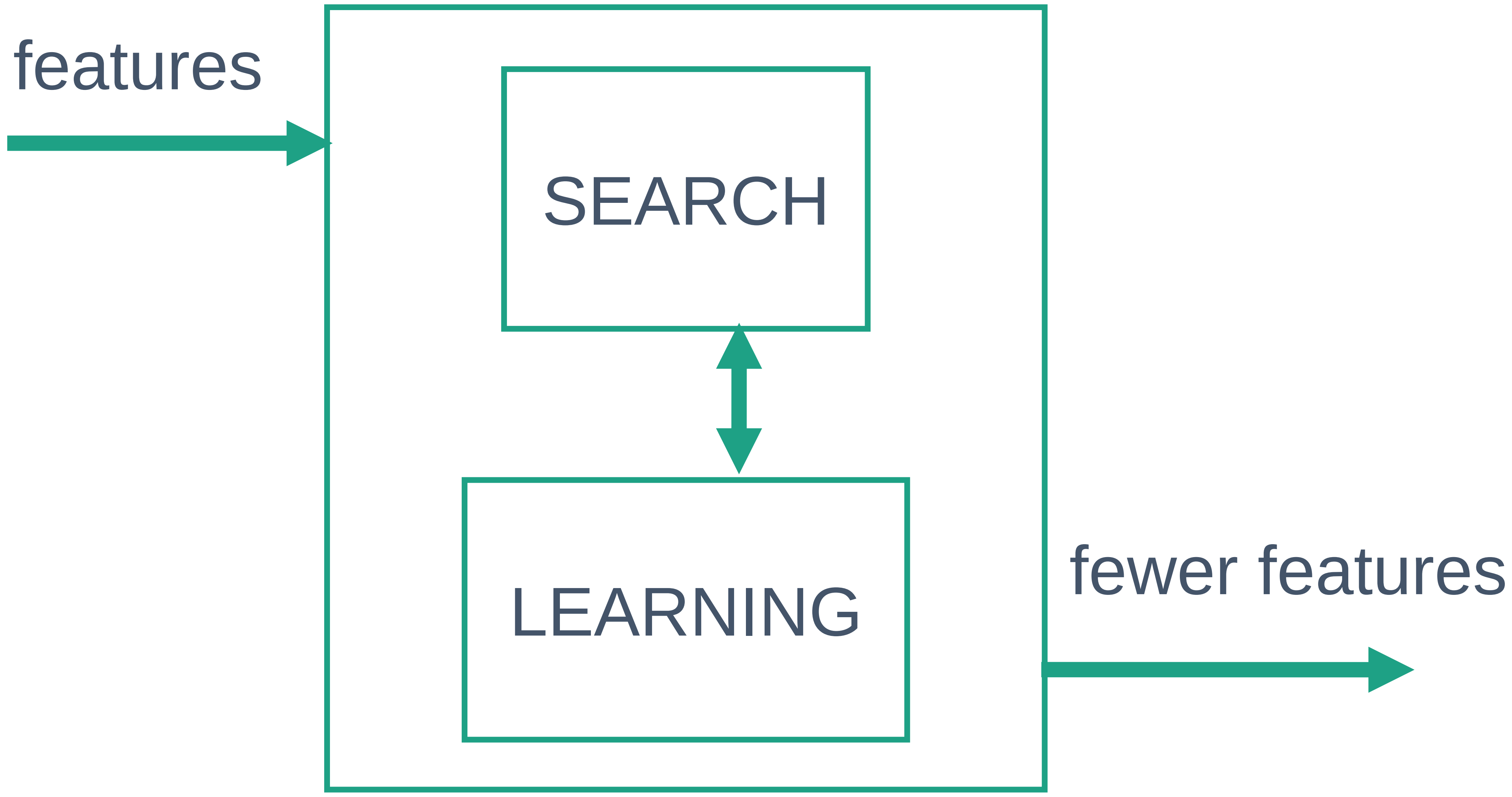
- The algorithm predicts whether the email is spam or not. Which feature is most useful for the prediction?
  - Feature 1: whether the email contains the word ‘viagra’
  - Feature 2: whether the email is sent from a Nigerian Prince
  - Feature 3: whether the email is sent from one person to a massive amount of people
- For all 1000 features you have calculated, maybe only a few features are important.
- Feature selection algorithm gives you insight and interpretability of your model.

Feature Selection

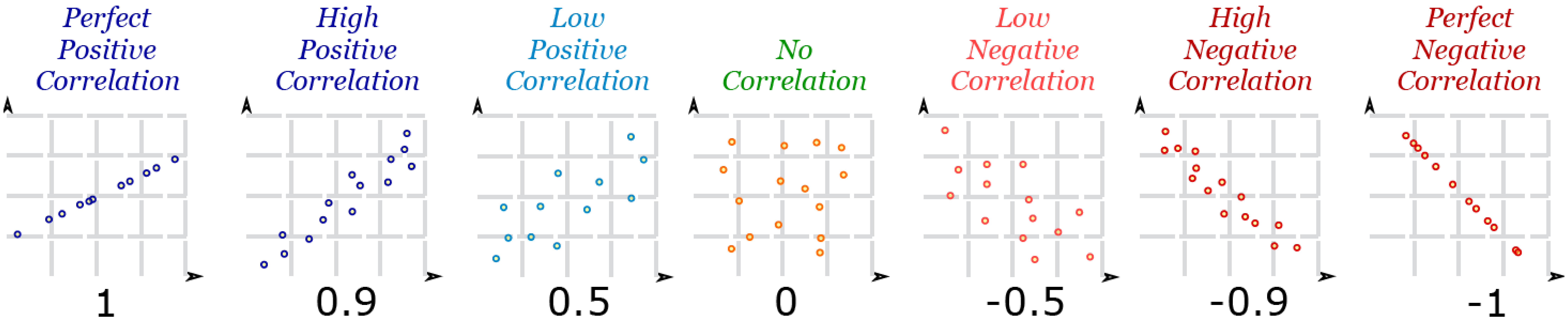
Filtering Methods



Wrapping Methods



Good Features?



- You need to find features that have high correlation to your target.
    - Don’t care whether it’s a positive or negative correlation
    - The larger the number the better

Good Features

- Correlation is not the only measure that tells you ‘how much x is related to y’ there are other measures we use.
- Such as:
  - ANOVA (Analysis of Variance)
  - Mutual Information
  - Chi2

Analysis of Variance (ANOVA)

- Classification problem: Y can only be class 0 or class 1. Find variance of X within class and between classes.

$$\text{F-Value} = \frac{\text{Variance between classes}}{\text{Variance within class}}$$

V between class is **high**
V within class is **low**
F-Value is **high**
Feature X is **important**

V between class is **low**
V within class is **high**
F-Value is **low**
Feature X is **not important**

Analysis of Variance (ANOVA)

Weight	Class
50	Adult
80	Adult
12	Children
30	Children
…	…

V between class is …
V within class is …
F-Value is …
Feature is …

Weight	Class
68	Thailand
75	China
80	Thailand
82	China
…	…

V between class is …
V within class is …
F-Value is …
Feature is …

Weight	Class
65	Human
1000	Animal
0.1	Animal
60	Animal
30	Human

V between class is …
V within class is …
F-Value is …
Feature is …

Feature with Low Variance

- A simple baseline approach to feature selection.
- It removes all features whose variance doesn’t meet some threshold.
- At the very least, we should remove all zero-variance features, i.e. features that have the same value in all samples.

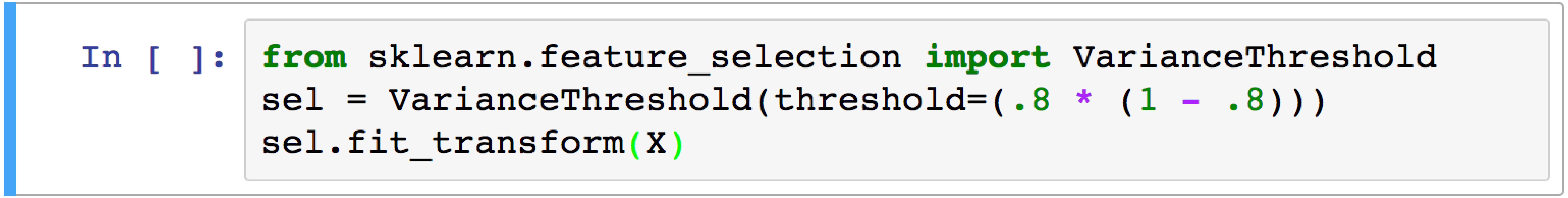
Low Variance Feature Removal

- A simple baseline approach to feature selection.
- It removes all features whose variance doesn’t meet some threshold.
- At the very least, we should remove all zero-variance features, i.e. features that have the same value in all samples.
- As an example, suppose that we have a dataset with boolean features, and we want to remove all features that are either one or zero (on or off) in more than 80% of the samples.
- Boolean features are Bernoulli random variables, and the variance of such variables is given by

$$Var[X] = p(1-p)$$

Low Variance Feature Removal

In sklearn, you may input the desired level of variance for VarianceThreshold selector.

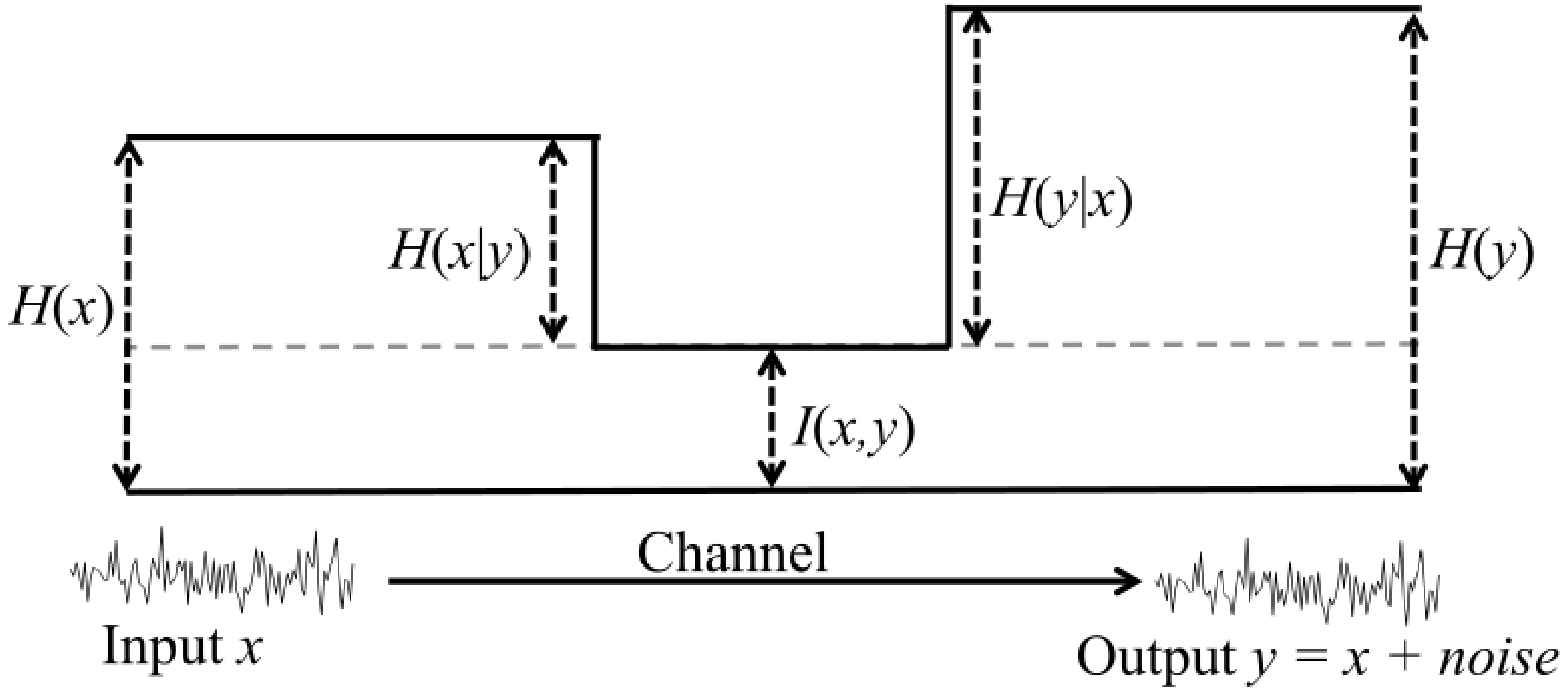


```
In [ ]: from sklearn.feature_selection import VarianceThreshold
        sel = VarianceThreshold(threshold=(.8 * (1 - .8)))
        sel.fit_transform(X)
```

- sklearn will automatically drop the columns where variance does not meet the desired level.
- Note that this can take care of one-hot encoded features with small number of ones.

Mutual Information

- Mutual Information I(X;Y) is the same as information gain, it is the information variable X and Y share or relevancy between two variables.
- Mutual information is symmetric, I(X;Y) is the same as I(Y;X).

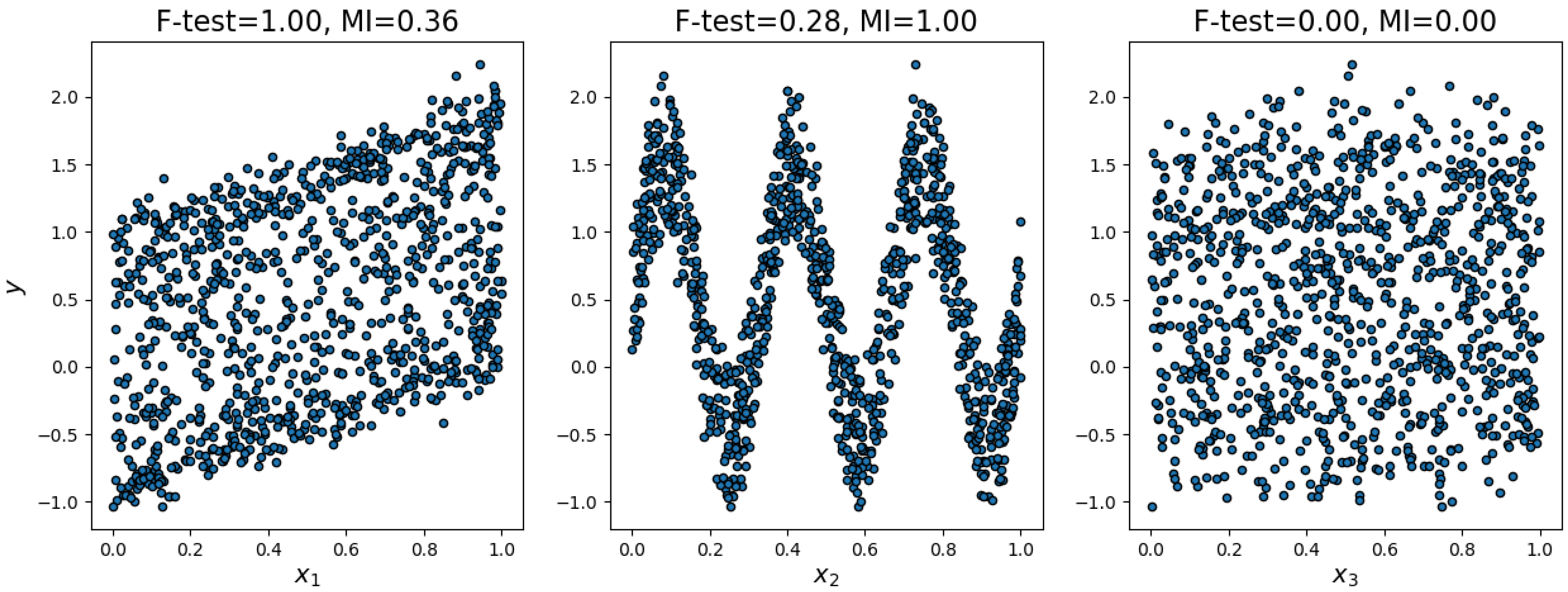


$$\begin{aligned} I(X;Y) &\equiv \mathrm{H}(X) - \mathrm{H}(X|Y) \\ &\equiv \mathrm{H}(Y) - \mathrm{H}(Y|X) \\ &\equiv \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y) \\ &\equiv \mathrm{H}(X,Y) - \mathrm{H}(X|Y) - \mathrm{H}(Y|X) \end{aligned}$$

$$H(X,Y) = -\sum_x \sum_y P(x,y) \log_2 [P(x,y)]$$

F-Test vs Mutual Information

- As F-test captures only linear dependency, it rates x1 as the most discriminative feature.
- On the other hand, mutual information can capture any kind of dependency between variables and it rates x2 as the most discriminative feature, which probably agrees better with our intuitive perception for this example.
- Both methods correctly marks x3 as irrelevant.



Mutual Information

- Mutual information methods can capture any kind of statistical dependency, but being nonparametric, they require more samples for accurate estimation.
- Mutual information is a non-negative value, the higher the more information shared between two variables.
- Can be calculated for both regression and classification problem.
  - sklearn mutual_info_regression: http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html
  - sklearn mutual_info_classif: http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html

Mutual Information

- For mutual information, you might need to identify each feature as discrete or continuous before proceeding.
- Unlike F-Test, mutual information does not have any p-value, so you may not know whether a particular value of MI is significant or not. They are used as feature ranking measure, not thresholded measure.

Chi-squared Test

- For two categorical variables with r classes and c classes respectively, you can make a contingency table that has r rows and c columns.
- The chi square test can be thought of as a test of independence. In a test of independence the null and alternative hypotheses are:
  - Ho: The two categorical variables are independent.
  - Ha: The two categorical variables are related.

http://math.hws.edu/javamath/ryan/ChiSquare.html

Chi-squared Test

- Example: we want to know whether continents (X) impact the frequency of different types of malaria (Y).

	Asia	Africa	South America	**Total**
Malaria A	31	14	45	**90**
Malaria B	2	5	53	**60**
Malaria C	53	45	2	**100**
Totals	**86**	**64**	**100**	**250**

Chi-squared Test

- We can use the equation to compute X^2

$$\chi_c^2 = \sum_i \frac{(FO_i - FE_i)^2}{FE_i}$$

	Asia	Africa	South America	**Total**
Malaria A	31	14	45	**90**
Malaria B	2	5	53	**60**
Malaria C	53	45	2	**100**
Totals	**86**	**64**	**100**	**250**

(90/250)*(86/250)*250

Observed	Expected	|O -E|	$(O — E)^2$	$(O — E)^2/$ E
31	30.96	0.04	0.0016	0.0000516
14	23.04	9.04	81.72	3.546
45	36.00	9.00	81	2.25
2	20.64	18.64	347.45	16.83
5	15.36	10.36	107.33	6.99
53	24.00	29.00	841	35.04
53	34.40	18.60	345.96	10.06
45	25.60	19.40	376.36	14.7
2	40.00	38.00	1444.00	36.1

Chi Square = 125.516

http://math.hws.edu/javamath/ryan/ChiSquare.html

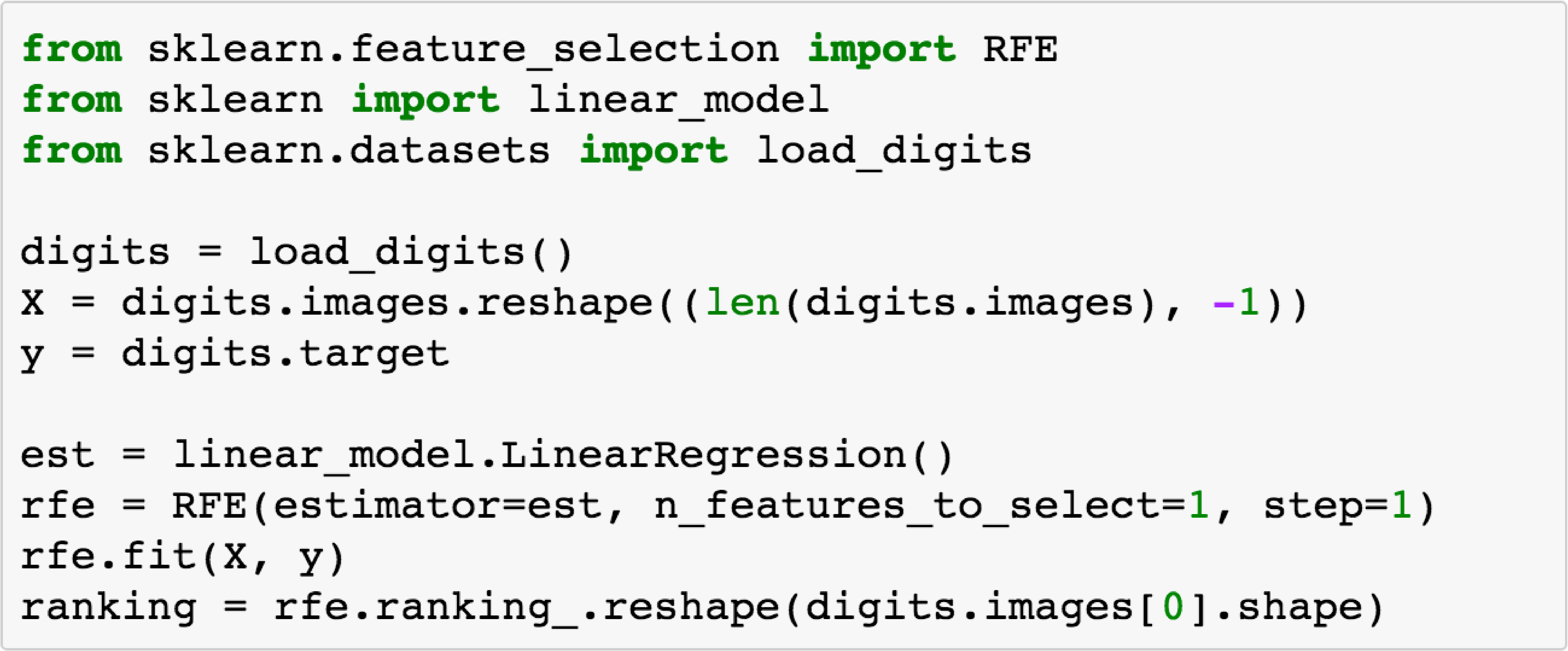
Chi-squared Test

- Similar to F-Test, Chi-Squares measure must be compared against a lookup table to determine p-value (statistical significance of the chi-square numbers).
- You can then use this p-value to assess importance of your features.
- Note that Chi-Squared works best when both features and targets are categorical. There is no continuous version. Features must be non-negatives, such as booleans or frequencies.
- more info: http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html

Wrapper Method

- Recursive Feature Elimination is an example of wrapper methods for feature selection.
- Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model)
- Recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features.
- First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a coef_ attribute or through a feature_importances_ attribute.
- Then, the least important features are pruned from current set of features.That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

Wrapper Method



```python
from sklearn.feature_selection import RFE
from sklearn import linear_model
from sklearn.datasets import load_digits

digits = load_digits()
X = digits.images.reshape((len(digits.images), -1))
y = digits.target

est = linear_model.LinearRegression()
rfe = RFE(estimator=est, n_features_to_select=1, step=1)
rfe.fit(X, y)
ranking = rfe.ranking_.reshape(digits.images[0].shape)
```

Wrapper Method

- It is important to consider feature selection a part of the model selection process. If you do not, it may lead to overfitting.
- Include feature selection within the inner-loop when you do cross-validation.
- A mistake would be to perform feature selection first to prepare your data, then perform model selection and training on the selected features.
- If you perform feature selection on all of the data and then cross-validate, then the test data in each fold of the cross-validation procedure was also used to choose the features and this is what biases the performance analysis.