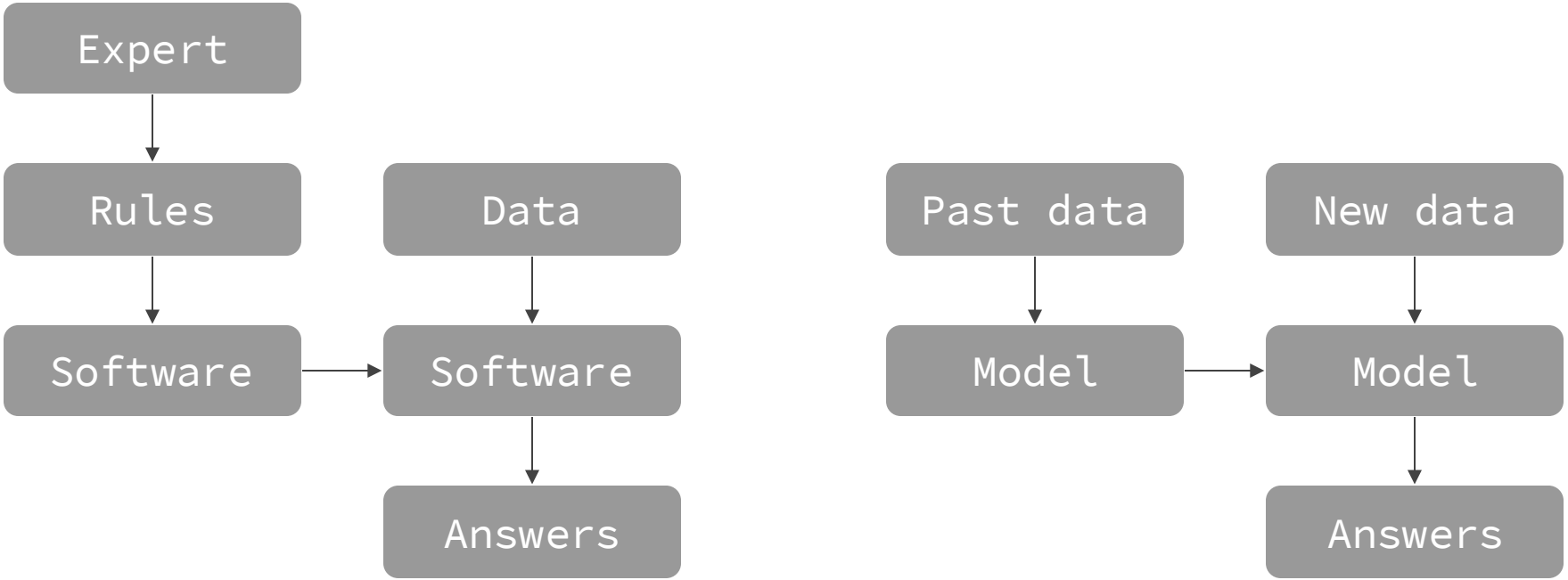
∂S Lab

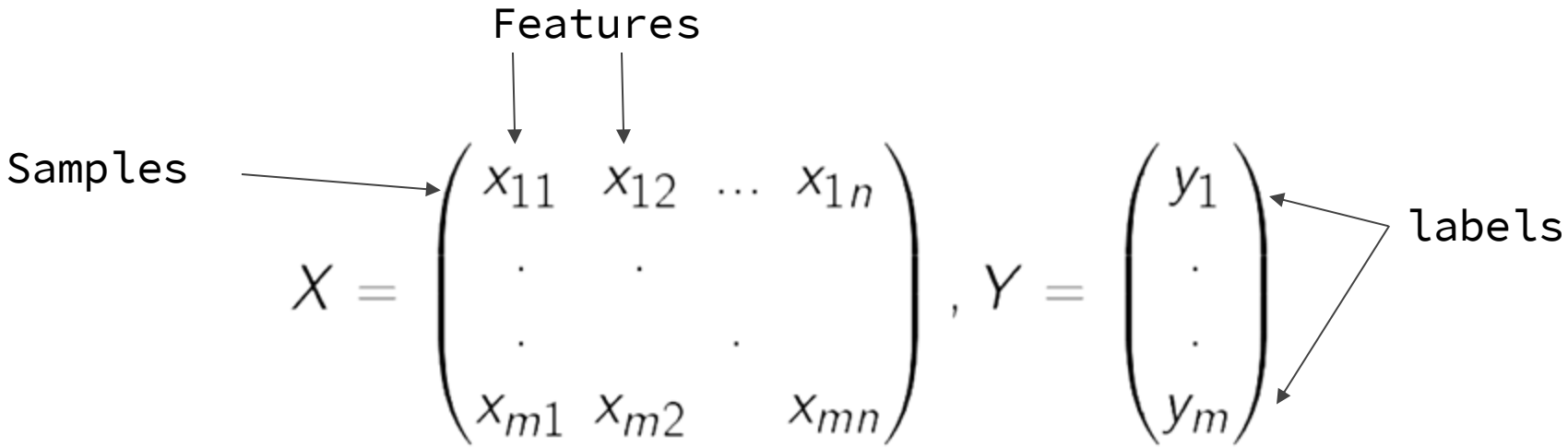
Lecture #6, Features

What is machine learning?

In general


Expert
Rules
Data
Software
Software
Answers
Past data
New data
Model
Model
Answers

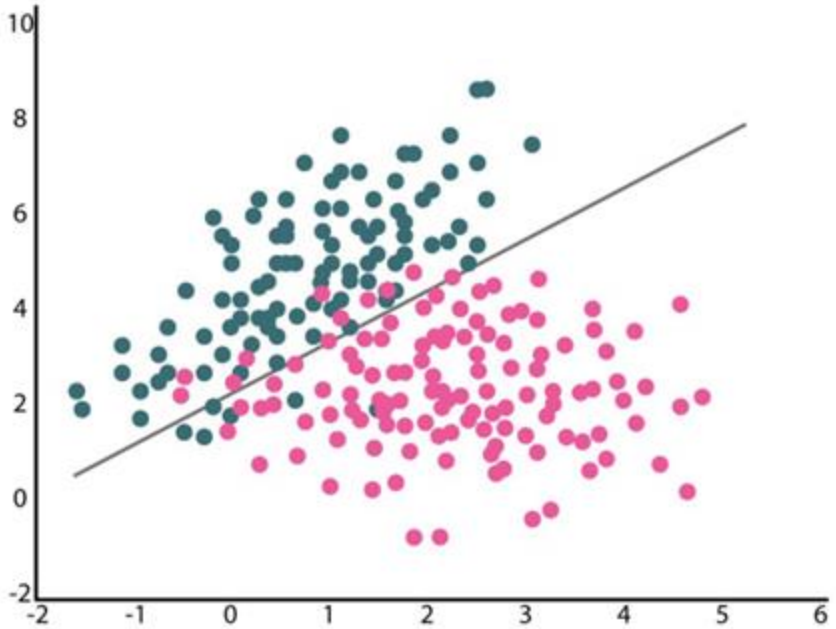
Definitions



Supervised learning. Binary classification

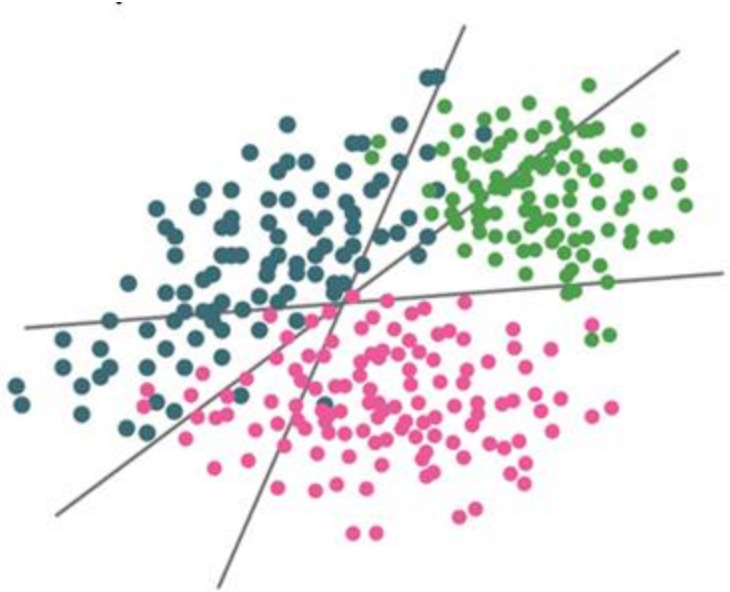
Y = {0, 1}

What could be X?



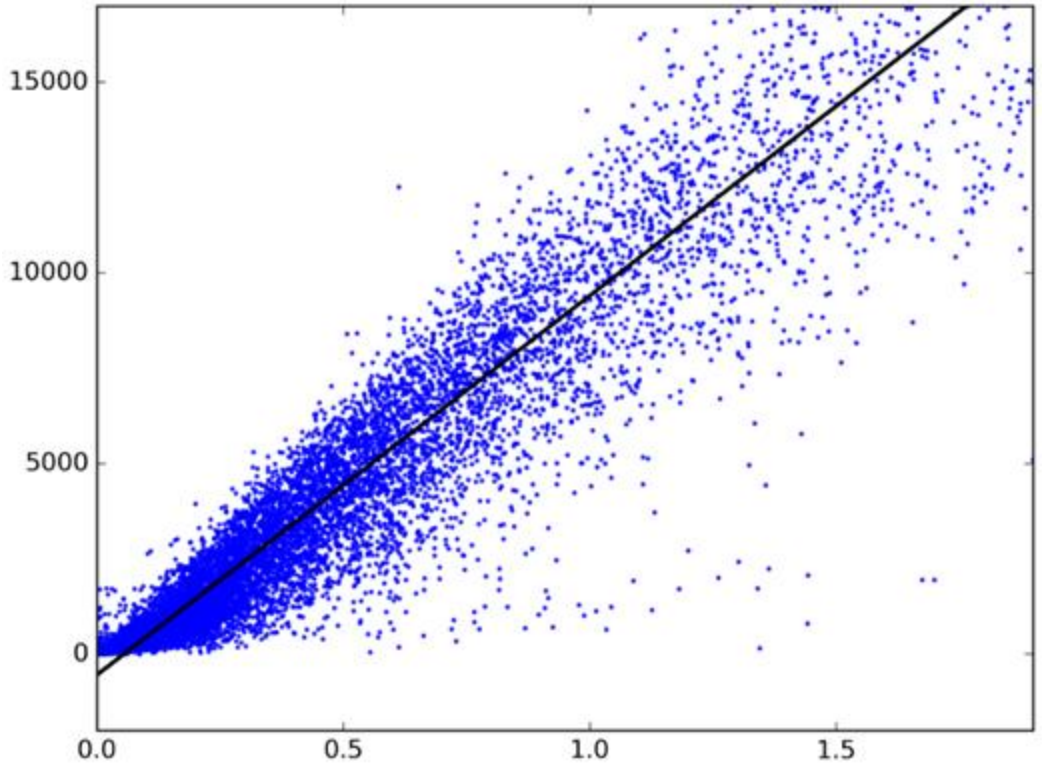
Supervised learning. Multiclass classification

$Y = \{0, 1, \ldots K\}$



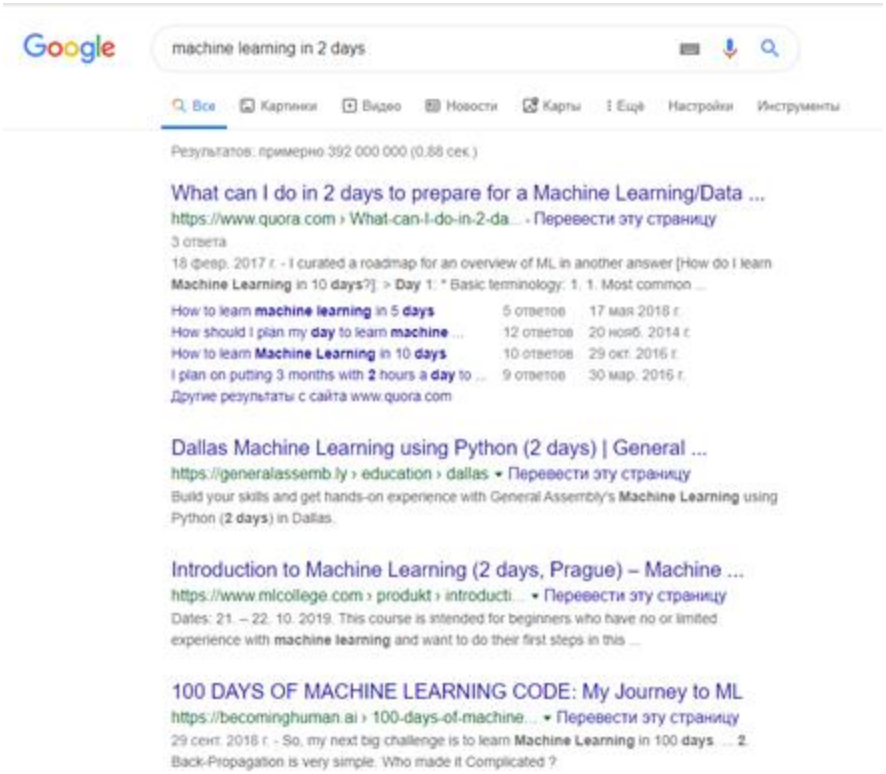
Supervised learning. Regression

Y = R



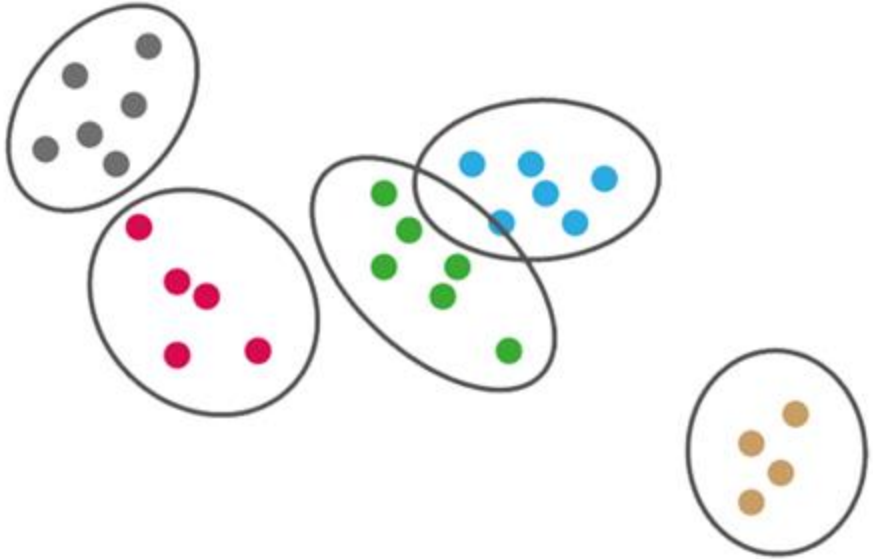
Supervised learning. Learning to rank

Only for students who choose this topic for capstone project :)



Unsupervised learning. Clusterization

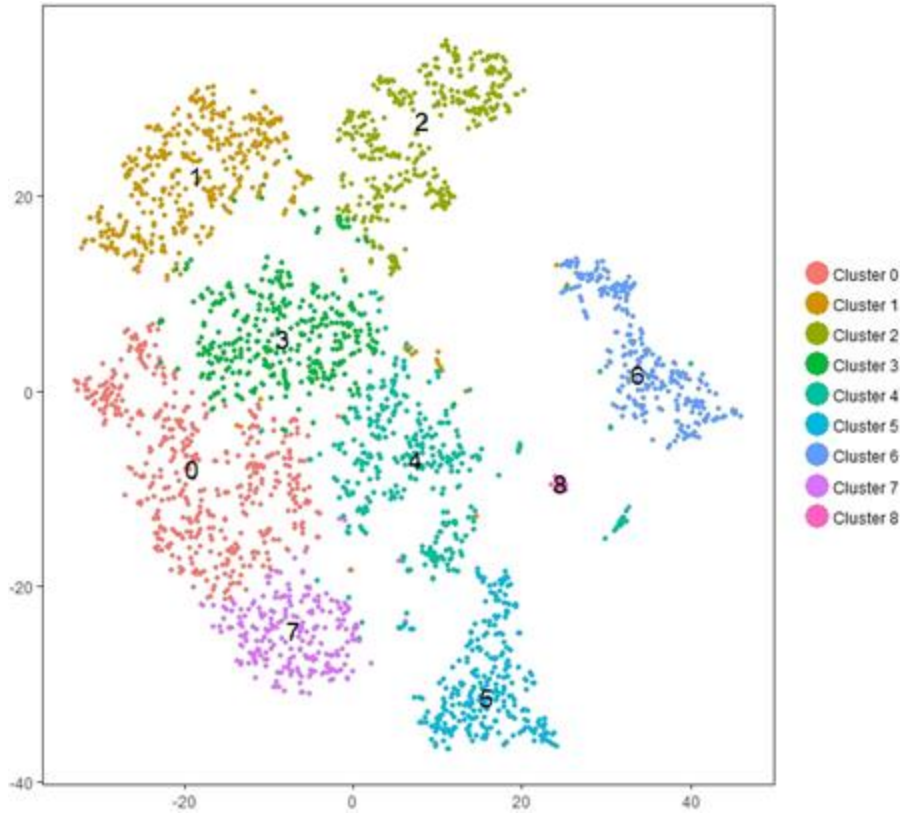
- Find groups of objects
- Don’t know labels
- Usually don’t know how many groups



Unsupervised learning. Visualization

Goals:

- Visualize multidimensional sampling
- It should have a structure
- It should be beautiful :)

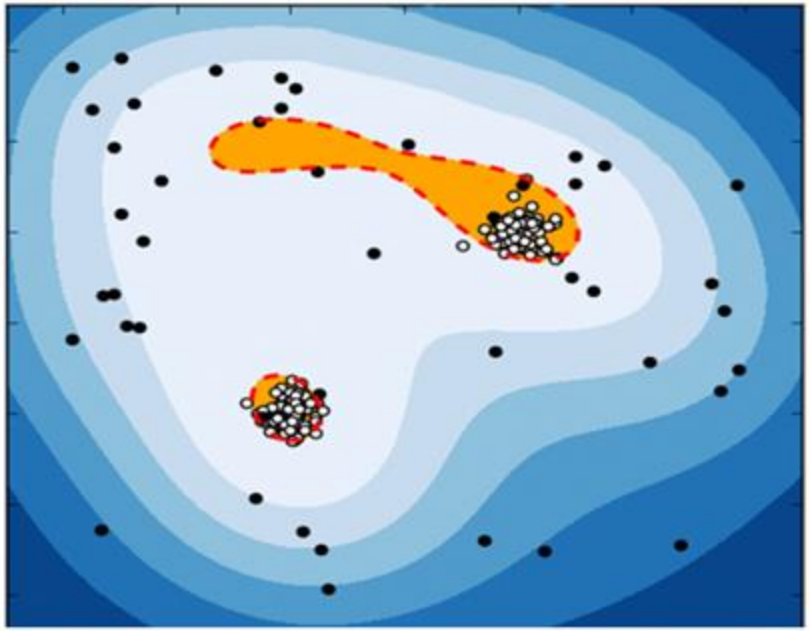


Unsupervised learning. Anomaly detection

Find VIP clients

Fraud detection

...



What features do we have?

Binary features

Real-valued features

Categorical features

Ordinal features

Correlations

Pearson

- Only for real-valued variables
- Unstable to outliers
- Only for linear relationship
- “Independent” ≠ “Uncorrelated”

Spearman

- Only for real-valued and ordinal variables
- Stable to outliers
- For monotone relationship

Non-real-valued cases

Binary	• The Matthews correlation
Categorical	• Cramér's V
Mixed	• Analysis of the corresponding expected values for each category

Missing records

What is wrong with them?

What could we do?

1. Analyse
2. Save all information (!)
3. Try different variants for imputation:
    a. Do nothing!
    b. Drop them
    c. Encode missing values with separate blank value (‘n/a’)
    d. Use the most probable value of the feature (mean, median, the most common)
    e. Encode with some extreme value
    f. Take the adjacent value – next or previous – for ordered data
    g. Build a special model :)

Scaling

sklearn.preprocessing.StandardScaler

Standardize features by removing the mean and scaling to unit variance.

Distribution becomes close to normal with mean=0 and std=1.

Scaler guarantees only for 68% of the data that it would be in [-1,1]

The standard score of a sample x is calculated as:

$$x = \frac{z - u}{s}$$

u - mean

s - standard deviation

sklearn.preprocessing.MinMaxScaler

Scale the feature to a given range [min, max]. Save information in the data (e.x. outliers!)

$$X_{scaled} = \frac{(max - min)}{X_{max} - X_{min}} * X + min - X_{min} * \frac{(max - min)}{X_{max} - X_{min}}$$

For [0,1]:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Categorical features

Label encoding

- Special labels for each category, like ordinal type of variables
- Value counts
- Any statistics of another feature calculated for each category (not TARGET)

One-hot encoding

Pros:
- Save information

Cons:
- Dimensional curse

color	color_red	color_blue	color_green
red	1	0	0
green	0	0	1
blue	0	1	0
red	1	0	0

Best practice:
use OneHotEncoder for features with not more than 15 unique categories.

k-Nearest Neighbors algorithm

How does it work?

```
sklearn.neighbors.

KNeighborsClassifier(

    n_neighbors=5,

    weights='uniform',

    metric='minkowski',

    p=2,

    n_jobs=None,

    …)
```