

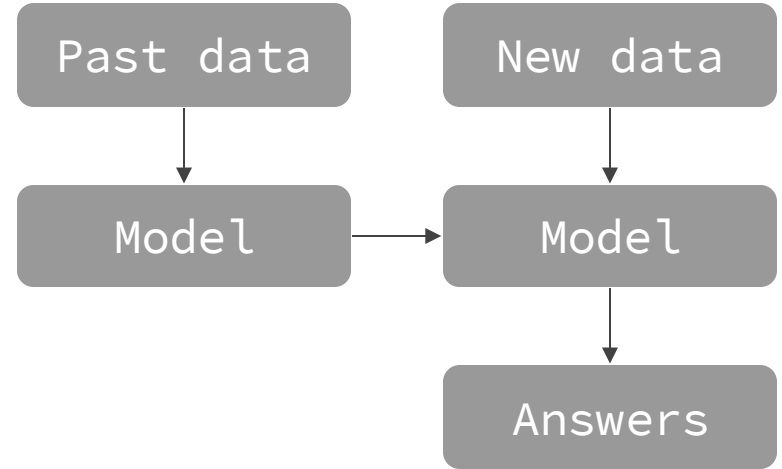
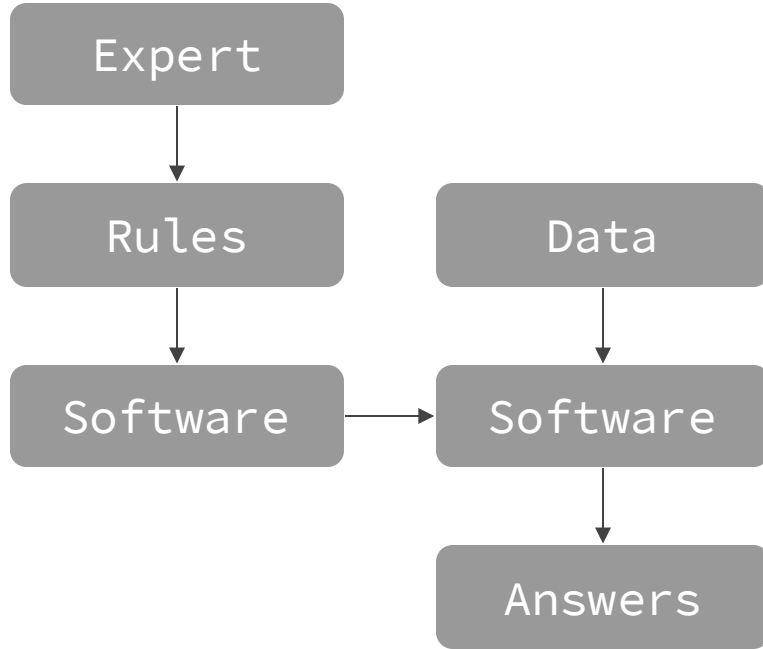
# DS Lab

## Lecture #6, Features

# What is machine learning?

# In general

---



# Definitions

Diagram illustrating the definitions of matrices  $X$  and  $Y$  in a machine learning context.

**Matrix  $X$  (Features matrix):**

The matrix  $X$  is defined as:

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ \vdots & \vdots & & \\ \vdots & & & \\ x_{m1} & x_{m2} & & x_{mn} \end{pmatrix}$$

Annotations for  $X$ :

- Samples:** An arrow points from the word "Samples" to the first column of the matrix.
- Features:** Two arrows point from the word "Features" to the first and second rows of the matrix.

**Matrix  $Y$  (Labels matrix):**

The matrix  $Y$  is defined as:

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

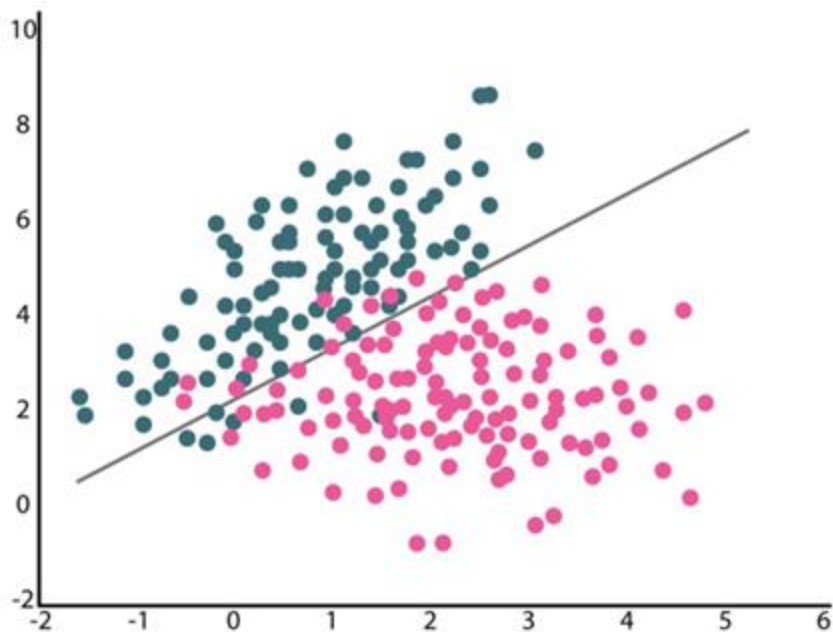
Annotation for  $Y$ :

- Labels:** An arrow points from the word "Labels" to the first row of the matrix.

# Supervised learning. Binary classification

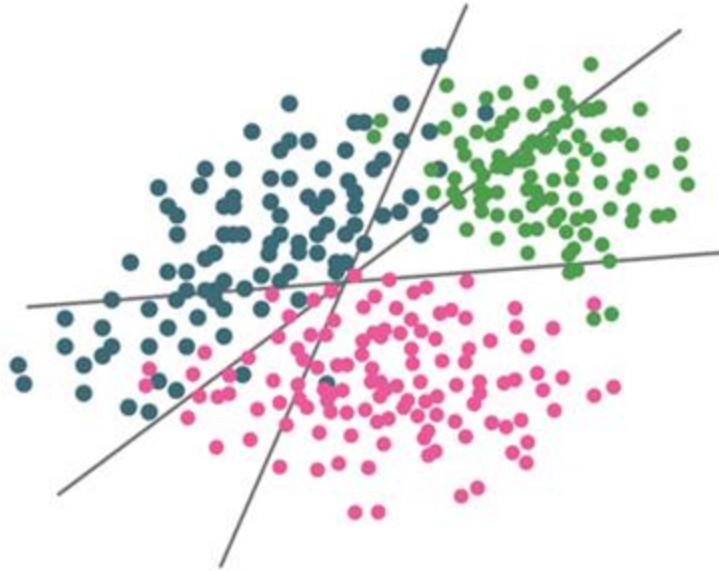
$Y = \{0, 1\}$

What could  
be  $X$ ?



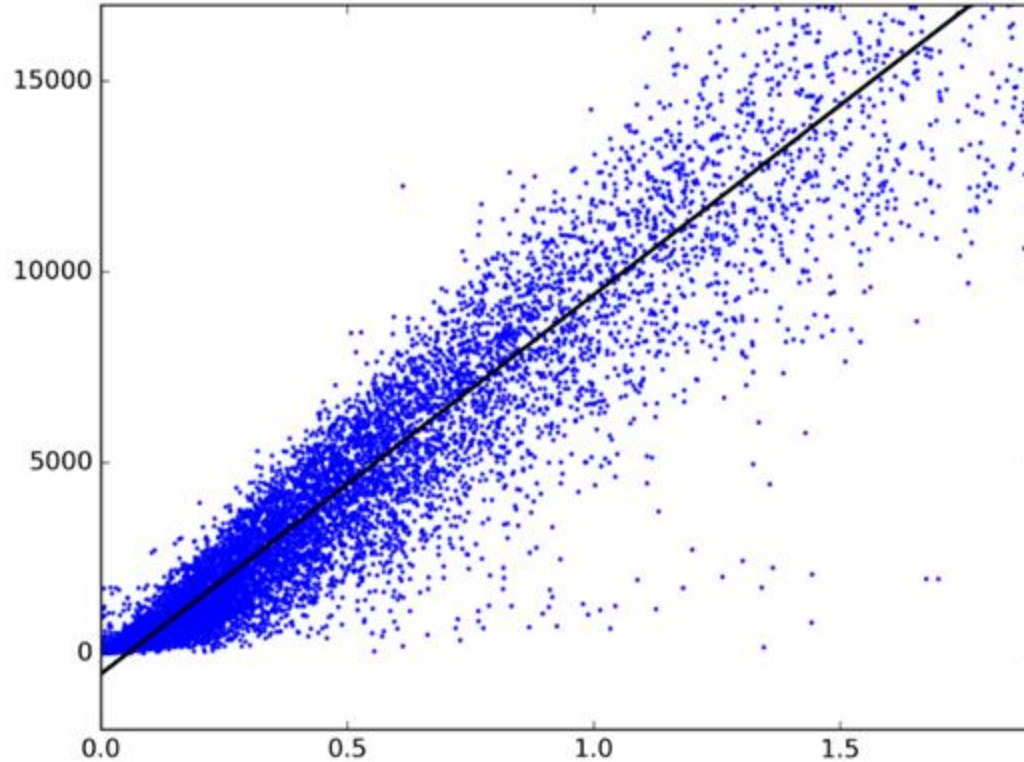
# Supervised learning. Multiclass classification

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



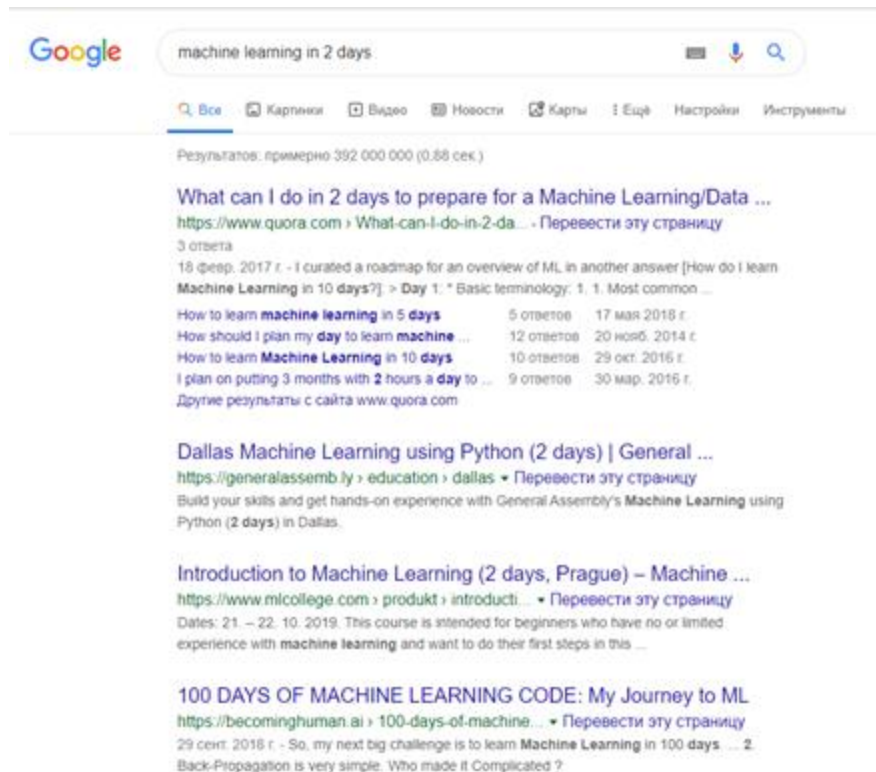
# Supervised learning. Regression

$$Y = R$$



# Supervised learning. Learning to rank

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



The image is a screenshot of a Google search results page for the query "machine learning in 2 days". The search bar at the top shows the query and the Google logo. Below the search bar, there are navigation tabs for "Все" (All), "Картинки" (Images), "Видео" (Videos), "Новости" (News), "Карты" (Maps), "Еще" (More), "Настройки" (Settings), and "Инструменты" (Tools). The search results show approximately 392,000,000 results in 0.68 seconds. The first result is a Quora link titled "What can I do in 2 days to prepare for a Machine Learning/Data ...". Below this, there is a table of related questions and answers. The second result is a link to "Dallas Machine Learning using Python (2 days) | General ...". The third result is a link to "Introduction to Machine Learning (2 days, Prague) – Machine ...". The fourth result is a link to "100 DAYS OF MACHINE LEARNING CODE: My Journey to ML ...".

Google machine learning in 2 days

Все Картинки Видео Новости Карты Еще Настройки Инструменты

Результатов: примерно 392 000 000 (0.68 сек.)

What can I do in 2 days to prepare for a Machine Learning/Data ...  
<https://www.quora.com/What-can-I-do-in-2-da...> • Перевести эту страницу  
3 ответа  
18 февр. 2017 г. - I curated a roadmap for an overview of ML in another answer [How do I learn Machine Learning in 10 days?]. > Day 1: \* Basic terminology: 1. 1. Most common ...

How to learn machine learning in 5 days	5 ответов	17 мая 2018 г.
How should I plan my day to learn machine ...	12 ответов	20 нояб. 2014 г.
How to learn Machine Learning in 10 days	10 ответов	29 окт. 2016 г.
I plan on putting 3 months with 2 hours a day to ...	9 ответов	30 мар. 2016 г.

Другие результаты с сайта www.quora.com

Dallas Machine Learning using Python (2 days) | General ...  
<https://generalassembly.com/education/dallas> • Перевести эту страницу  
Build your skills and get hands-on experience with General Assembly's Machine Learning using Python (2 days) in Dallas.

Introduction to Machine Learning (2 days, Prague) – Machine ...  
<https://www.miccollege.com/product/introducti...> • Перевести эту страницу  
Dates: 21. – 22. 10. 2019. This course is intended for beginners who have no or limited experience with machine learning and want to do their first steps in this ...

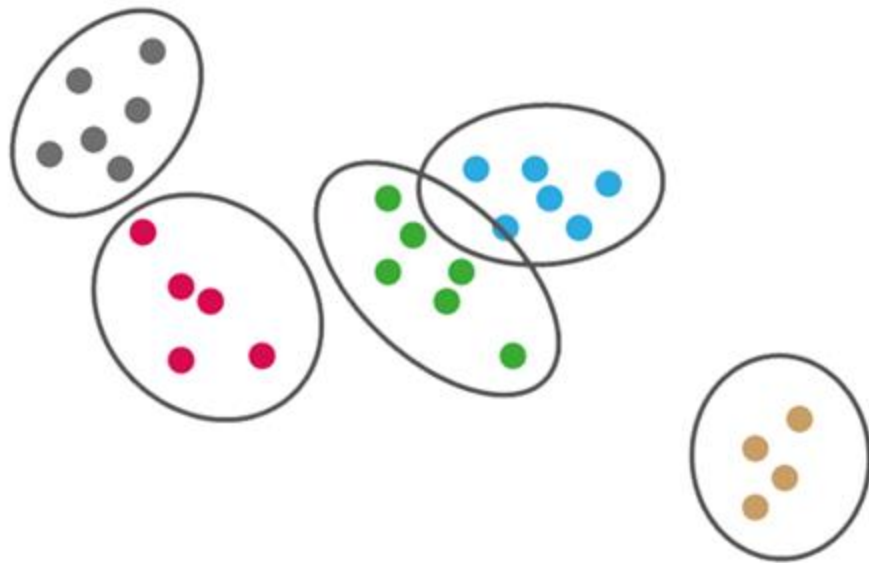
100 DAYS OF MACHINE LEARNING CODE: My Journey to ML  
<https://becominghuman.ai/100-days-of-machine...> • Перевести эту страницу  
29 сент. 2018 г. - So, my next big challenge is to learn Machine Learning in 100 days ... 2. Back-Propagation is very simple. Who made it Complicated ?



# Unsupervised learning. Clusterization

---

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

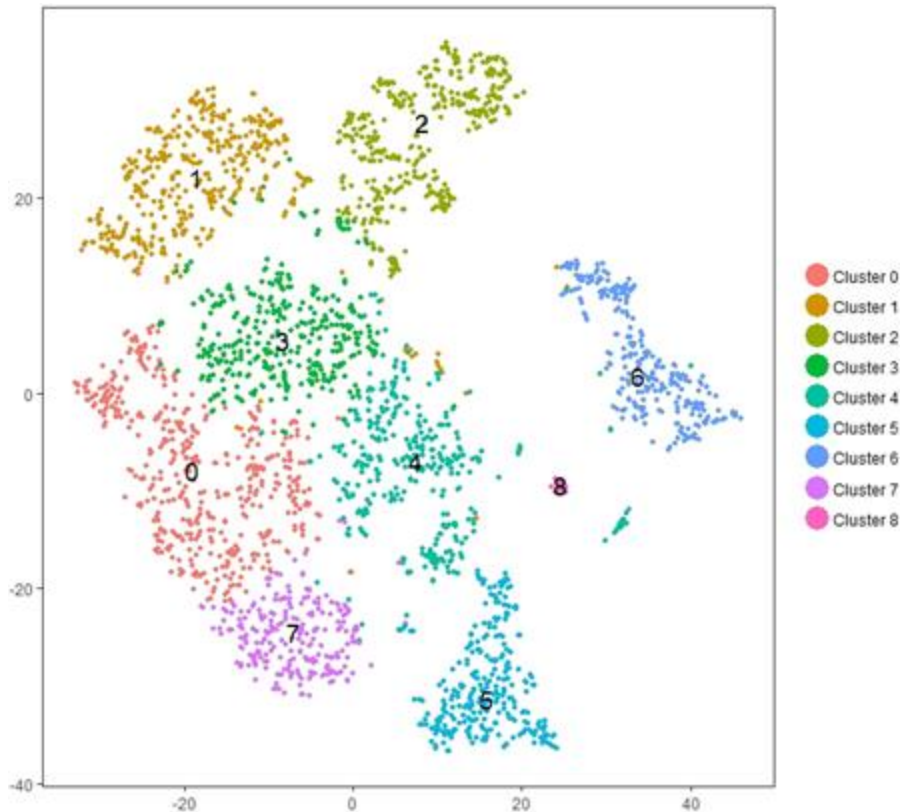


# Unsupervised learning. Visualization

---

Goals:

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



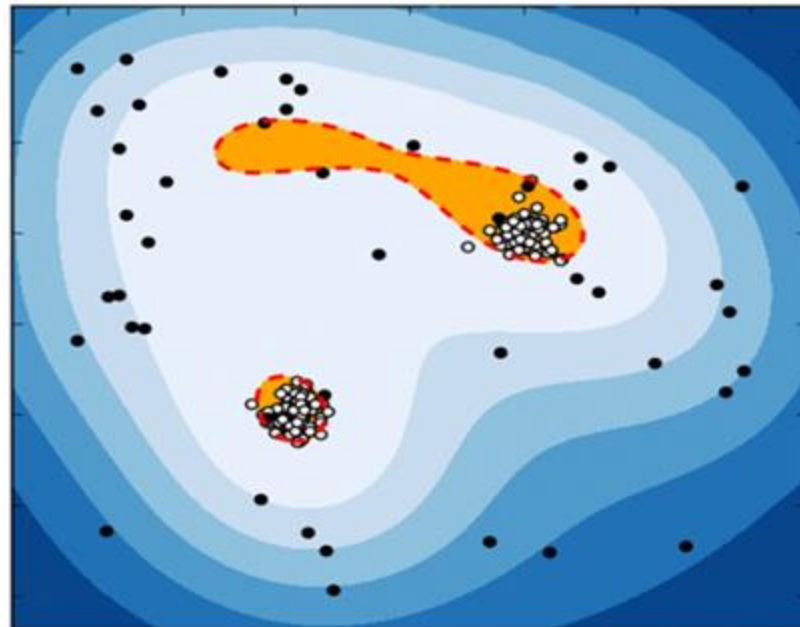
# Unsupervised learning. Anomaly detection

---

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”  $\neq$  “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,`

`...)`

