



AI

ITAM AI Lab MISIS

Лекция 1

**Что такое машинное обучение?
Соревнования по машинному обучению.
Основные термины, литература и источники.**



AI - Иванов Арсений

- 3 курс Прикладная математика
- сотрудник SberRecSys - ZVUK
- занимаюсь research'ем в SberAI

Изучаю и работаю в GNN, RecSys

Libs: Pandas, NumPy, SciPy, PyTorch, PyG, DgL

Мой TG-канал





Коммуникации



Сайт MISIS AI Lab

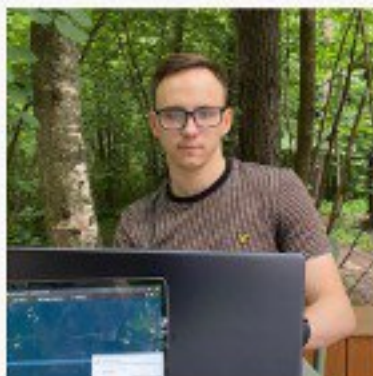


Чат в TG



TG-канал

Кто мы? И чем мы занимаемся?



**Изучение ML, Q/A,
общение**

**Data-Science
чемпионаты**

Хакатоны

**Research и
публикация статей**

Artificial intelligence

Machine Learning, Data Science



Data-Science
чемпионаты

DS-competitions



1. Работа с данными
2. Приближенные к рабочим задачи
3. Leader-board, math, ds

kaggle



Alcrowd



≡

kaggle

+

Create

🏠

Home

🏆

Competitions

📊

Datasets

⌕

Code

💬

Discussions

🎓

Learn

⌵


More

📁


Your Work

▼

RECENTLY VIEWED



H&M Personalized Fas...



Spaceship Titanic

📅

View Active Events

🔍

Search

👤

🏠

GettingStarted Prediction Competition

Spaceship Titanic

Predict which passengers are transported to an alternate dimension

k

Kaggle · 2,085 teams · Ongoing

Overview

Data

Code

Discussion

Leaderboard

Rules

Join Competition

...

Overview

Description

Evaluation

Frequently Asked Questions

💡


Recommended Competition

We highly recommend [Titanic - Machine Learning from Disaster](#) to get familiar with the basics of machine learning and Kaggle competitions.

Welcome to the year 2912, where your data science skills are needed to solve a cosmic mystery. We've received a transmission from four lightyears away and things aren't looking good.

The *Spaceship Titanic* was an interstellar passenger liner launched a month ago. With almost 13,000 passengers on board, the vessel set out on its maiden voyage transporting emigrants from our solar system to three newly habitable exoplanets orbiting nearby stars.

While rounding Alpha Centauri en route to its first destination—the torrid 55 Cancri E—the unwary *Spaceship Titanic* collided with a spacetime anomaly hidden within a dust cloud. Sadly, it met a similar fate as its namesake from 1000 years before. Though the ship stayed intact, almost half of the passengers were transported to an alternate dimension!





+ Create

Home

Competitions

Datasets

<> Code

Discussions

Learn

More

Your Work

View Active Events

Search

GettingStarted Prediction Competition

Spaceship Titanic

Predict which passengers are transported to an alternate dimension

Kaggle · 2,085 teams · Ongoing

Overview Data Code Discussion Leaderboard Rules

Join Competition

Leaderboard

Raw Data

Refresh

Search leaderboard

This leaderboard is calculated with all of the test data.

#	Team	Members	Score	Entries	Last	Code	Join
1	CADang		0.87304	28	15d		
2	Setec Astronomy		0.86626	136	21d		
3	Involucional Neural Network		0.84428	159	2d		
4	Karl Cini		0.82253	18	1mo		
5	Matan Peleg		0.81716	12	3d		

DS-competitions Right Now



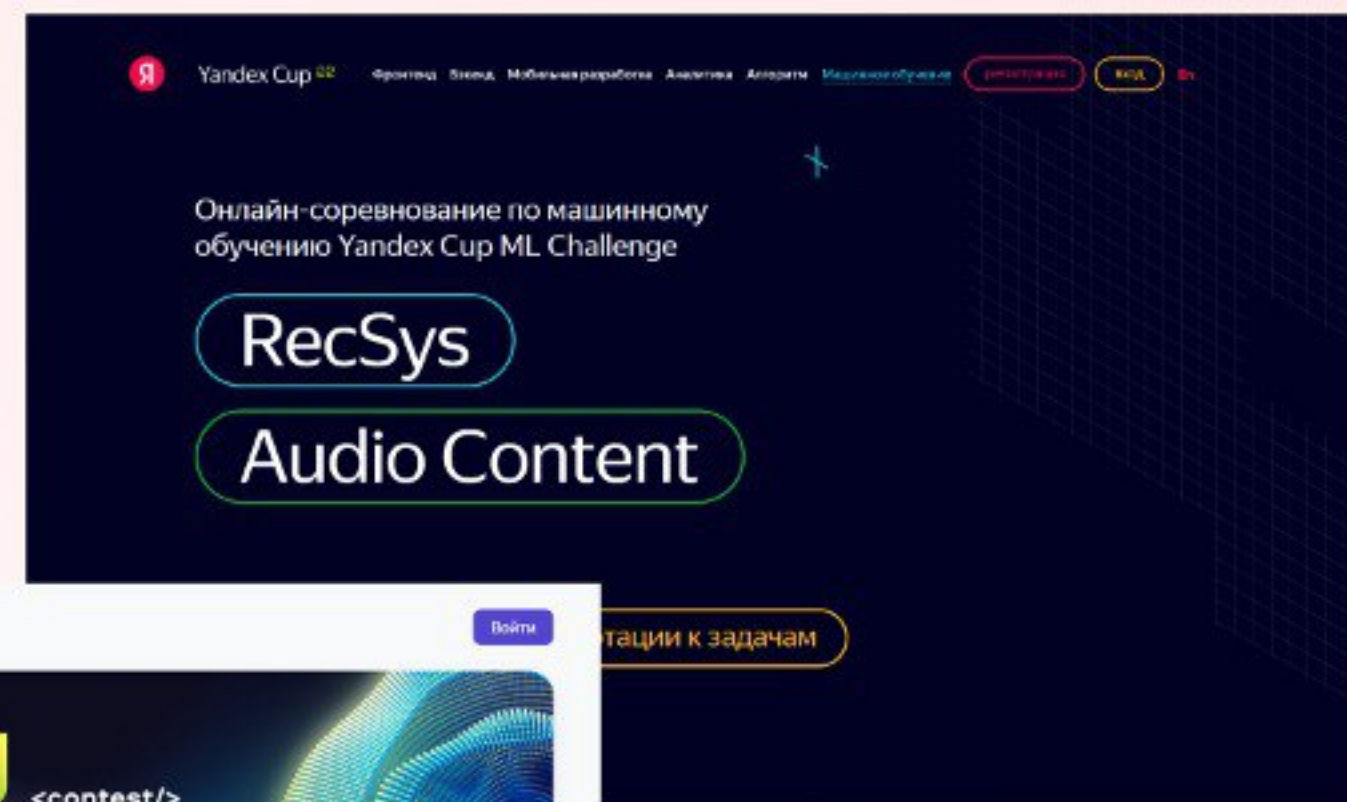
Task description

The Visual Question Answering task is simple for humans but difficult for machines. Given an image and a question, your mission is to build a system that draws a bounding box to answer the question.

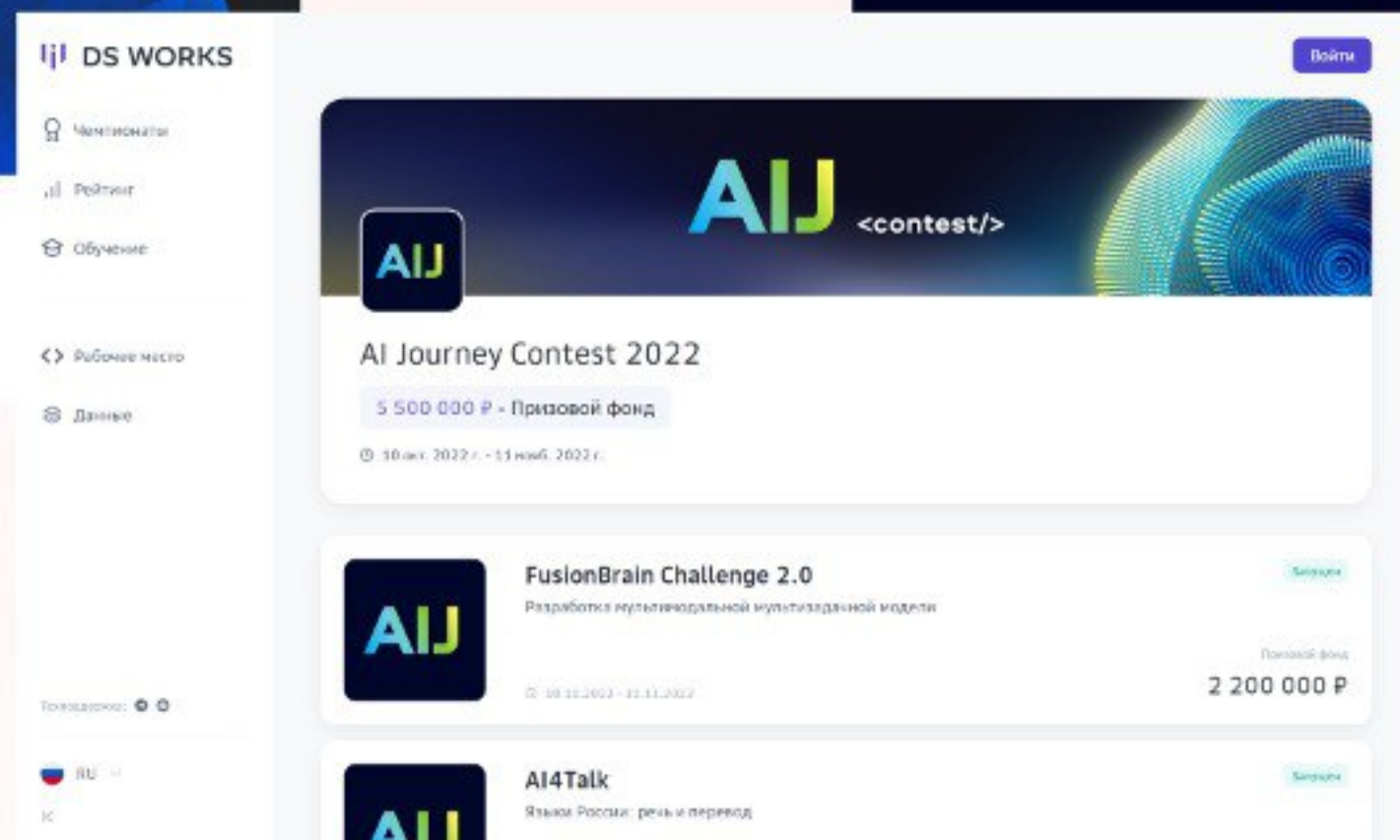
Examples

TOLOKA

AIJ



YaCup



Artificial intelligence

Machine Learning, Data Science



Hackatons



WHOOSH
MOBILITY
HACK

09-10 ОКТЯБРЯ 2021

ЭНЕРГИЯ
ПРОРЫВА

ПРИЗОВОЙ ФОНД
1 000 000 ₽

ГЛОБАЛЬНАЯ
ЭНЕРГИЯ



ЦИФРОВОЙ
ПРОРЫВ



Генеральный спонсор



РоссельхозБанк

AgroCode
Hack 2022

СБЕР ЗВУК

< Хакатон для python-разработчиков middle и senior уровня >

SberZvuk Tech Days

Призовой фонд 500 000 ₽ >

SberZvuk Tech Days

Призовой фонд 500 000 ₽ >

SberZvuk Tech Days

Призовой фонд 500 000 ₽ >

SberZvuk Tech Days

Призовой фонд 500 000 ₽ >

30-31 окт

30-31 окт

30-31 окт

ОНЛАЙН-ХАКАТОН

MOSCOW
CITY HACK

Создай новые цифровые продукты
и сервисы для города

Hackatons

Agora-Hack



TFIDF + SVD + KNN

На референсах - фаворитный метод обучения модели

Тренировка = 230мс+3сек на SVD
Векторизация = 1 сек

Плюсы:

- быстрый инференс
- нужно тренировать только на эталонах
- меньшая размерность векторов
- занимает мало места
- модель устойчива к выбросам

0.963

ACCURACY

Использование данной конфигурации модели показывает значительно высокую оценку



БЫСТРАЯ ВЕКТОРИЗАЦИЯ

Легковесный алгоритм и быстрое обучение модели



АВТОМАТИЗАЦИЯ

Оптимально и автоматизировано, поскольку в TFIDF используются sparse-матрицы

“Разработка системы с web-интерфейсом для сопоставления характеристик товаров маркетплейса с их эталонными значениями”



Artificial intelligence Machine Learning, Data Science



Research and science

Deformable Graph Convolutional Networks

Jinyoung Park, Sungdong Yoo, Jihwan Park, Hyunwoo J. Kim*

Department of Computer Science and Engineering, Korea University
{jyoun678, ysd424, jwpark7071, hyunwoo@kum.ac.kr}

Abstract

Graph neural networks (GNNs) have significantly improved the representation power for graph-structured data. Despite of the recent success of GNNs, the graph convolution in most GNNs have two limitations. Since the graph convolution is performed in a small local neighborhood on the input graph, it is inherently incapable to capture long-range dependencies between distant nodes. In addition, when a node has neighbors that belong to different classes, i.e., heterophily, the aggregated messages from them often negatively affect representation learning. To address the two common problems of graph convolution, in this paper, we propose Deformable Graph Convolutional Networks (Deformable GCNs) that adaptively perform convolution in multiple latent spaces and capture short/long range dependencies between nodes. Separated from node representations (features), our framework simultaneously learns the node positional embeddings (coordinates) to determine the relations between nodes in an end-to-end fashion. Depending on node position, the convolution kernels are deformed by deformation vectors and apply different transformations to its neighbor nodes. Our extensive experiments demonstrate that Deformable GCNs flexibly handles the heterophily and achieve the best performance in node classification tasks on six heterophilic graph datasets.

1 Introduction

Graphs are flexible representations for modeling relations in data analysis problems and are widely used in various domains such as social network analysis (Wang, Cui, and Zhu 2016), recommender system (Berg, Kipf, and Welling 2017), chemistry (Gilmer et al. 2017), natural language processing (Erxen and Raden 2004), and computer vision (Johnson et al. 2015). In recent years, Graph Neural Networks (GNNs) have achieved great success in many graph-related applications such as node classification (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2018), link prediction (Zhang and Chen 2018; Schlichtkrull et al. 2018), and graph classification (Erxen et al. 2019; Ying et al. 2018). Most existing GNNs learn node representations via message passing schemes, which iteratively learn

the hidden representation of each node by aggregating messages from its local neighborhoods. For example, Graph Convolution Networks (GCNs) (Kipf and Welling 2017) operate convolutions on input graphs inspired by first-order approximation of spectral graph convolutions (Hammond, Vandergheynst, and Gibouval 2011).

However, most graph convolution that aggregates messages from local neighborhoods implicitly assumes that input graphs are homophilic graphs, where connected nodes have similar features or belong to the same class. So the smoothing over the input graphs effectively removes noise in the input features and significant

tational power when the assumption is violated. In contrast, heterophilic graphs where connected nodes have different labels, the conventional neural networks often underperform. To address these limitations, we propose Deformable Graph Convolutional Networks (Deformable GCNs) that adaptively perform convolution in multiple latent spaces. Started from the discrete convolution with first-order approximation (Dai et al. 2020), we define a grid space for input graph and generate different transformations. Our framework models useful relations between nodes by the difference of latent embeddings. Our contributions are as follows:

- We propose a Deformable Graph Convolution (Deformable GCConv) that perform space and adaptively deforms to handle heterophily and variable latent nodes.
- We propose novel architecture for Deformable GCNs.

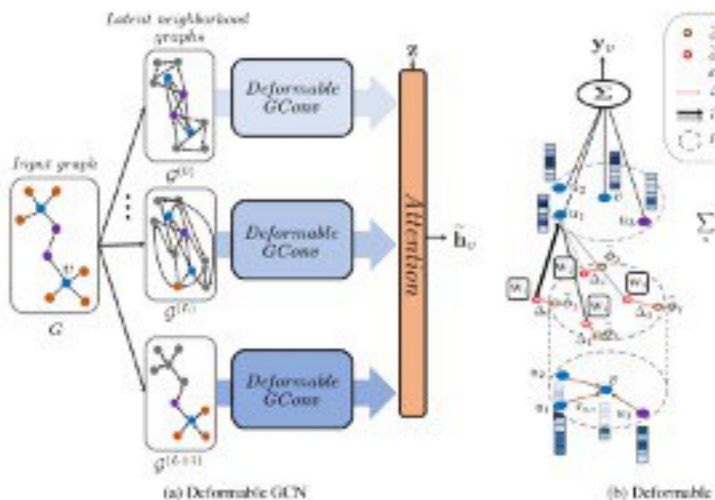


Figure 1: Overall structure of Deformable GCN and Deformable GCConv. (a) In Deformable GCN, at each layer, latent neighborhood graphs, $\{G^{(l)}\}_{l=0}^{L-1}$, are constructed to define neighborhoods for the Deformable Graph Convolution (Deformable GCConv). Then, Deformable GCConv is applied on each latent neighborhood graph and the outputs of the convolution $\{y_v^{(l)}\}_l$ are adaptively aggregated for representing h_v using an attention mechanism. (b) Our Deformable GCConv performs graph convolution in a latent (position) space. For more flexible graph convolution, Deformable GCConv adaptively deforms convolution kernels $g_{\text{kernel}}(\cdot, \cdot)$ for each center node v by kernel vector deformation $\Delta_v(\phi_v)$.

networks have difficulty adapting to graphs that linked nodes often have different properties, called heterophilic graphs.

In our work, we consider both heterophilic and homophilic graphs, unlike standard graph neural networks that have mainly focused on homophilic graphs. To have enough representation power on heterophilic graphs, we generate latent graphs for linking distant nodes with similar property according to their latent embeddings.

3.2 Deformable Graph Convolution

We here introduce a Deformable Graph Convolution (Deformable GCConv), which softly changes the receptive fields and adaptively aggregates messages from neighbors on the multiple latent graphs. The overall structure of Deformable

relation between v and u , $\tilde{N}(v)$ is the neighborhood of v that coincides with the finite support of g centered at v , $g(r_{u,v})$ is a linear function to transform h_u and it varies depending on the relation vector. For example, in a 2D convolution with a 3×3 kernel, $r_{u,v} = \phi_u - \phi_v$, where ϕ_u, ϕ_v are the coordinates of u and v . For each relative position, a linear function $g(r_{u,v}) \in \mathbb{R}^{k \times k}$ is applied.

In the graph domain, a GCN layer defined in (2) (without the activation function) can be viewed as a specific instantiation of (3) with $g(r_{u,v}) = (\deg(v)\deg(u))^{-1/2}W$. So, the GCN relations are determined by the degree of node u and v . Also, except for the normalization, GCN performs the same linear transformation for all the relations different

Dataset	Heterophilic Graphs						Homophilic Graphs		
	Texas	Wisconsin	Actor	Squirrel	Chameleon	Cornell	Citeseer	Pubmed	Cora
# Classes	5	5	5	5	5	5	6	3	7
# Nodes	183	251	7600	5201	2277	183	3327	19717	2708
# Edges	279	450	26659	198353	31371	277	4552	44324	5278
# Features	1701	1703	932	2089	2325	1703	3703	500	1433
Avg deg	3.05	3.59	7.02	76.28	27.55	3.05	3.05	4.50	3.90
Hom. ratio \hat{h}	0.11	0.21	0.22	0.22	0.23	0.30	0.74	0.80	0.81

Table 1: Dataset statistics.

Dataset	Texas	Wisconsin	Actor	Squirrel	Chameleon	Cornell	Citeseer	Pubmed	Cora
Hom. ratio \hat{h}	0.11	0.21	0.22	0.22	0.23	0.30	0.74	0.80	0.81
MLP	82.16 \pm 2.64	84.90 \pm 1.82	36.78 \pm 0.26	30.77 \pm 1.18	47.87 \pm 1.21	81.08 \pm 0.62	72.86 \pm 1.42	87.62 \pm 0.29	75.09 \pm 1.45
GCN	64.32 \pm 2.40	62.94 \pm 1.15	30.47 \pm 0.64	46.65 \pm 0.99	64.98 \pm 0.98	60.27 \pm 2.37	76.66 \pm 1.32	87.59 \pm 0.38	87.44 \pm 0.35
GAT	60.81 \pm 0.82	64.31 \pm 0.38	29.92 \pm 0.43	45.47 \pm 1.18	66.56 \pm 0.98	59.46 \pm 1.38	76.54 \pm 1.20	86.55 \pm 0.34	87.46 \pm 0.37
ChenNet	76.49 \pm 1.45	77.84 \pm 0.28	35.01 \pm 0.75	45.42 \pm 0.66	61.80 \pm 1.08	72.97 \pm 0.44	76.20 \pm 1.12	89.16 \pm 0.30	83.44 \pm 0.64
JKNet	62.97 \pm 1.15	60.78 \pm 0.29	30.78 \pm 0.80	53.78 \pm 1.25	68.53 \pm 1.90	59.19 \pm 1.15	76.05 \pm 1.00	88.64 \pm 0.33	87.17 \pm 0.83
Mishop	83.78 \pm 0.35	85.10 \pm 0.25	33.80 \pm 0.89	39.32 \pm 0.16	63.09 \pm 0.16	63.09 \pm 0.16	76.05 \pm 1.00	88.64 \pm 0.33	87.17 \pm 0.83
Geom-GCN	68.11 \pm 1.04	64.51 \pm 0.35	31.48 \pm 0.64	38.00 \pm 0.80	60.99 \pm 0.16	60.99 \pm 0.16	76.05 \pm 1.00	88.64 \pm 0.33	87.17 \pm 0.83
H ₂ GCN	82.16 \pm 0.12	86.67 \pm 0.18	36.96 \pm 0.33	54.51 \pm 0.56	65.42 \pm 0.16	65.42 \pm 0.16	76.05 \pm 1.00	88.64 \pm 0.33	87.17 \pm 0.83
Deformable GCN	84.32\pm1.34	87.66\pm1.16	37.07\pm0.79	62.56\pm1.21	70.90\pm0.16	70.90\pm0.16	76.05\pm1.00	88.64\pm0.33	87.17\pm0.83

Table 2: Evaluation results on node classification task (Mean accuracy (%)). models are highlighted with boldface.

is reported with the best model on the validation datasets. For all datasets, we apply the splits (48%/32%/20%)¹ of nodes per class for (train/validation/test) provided by (Pei et al. 2020) for a fair comparison as (Zhu et al. 2020). All experiments are repeated 10 times as (Pei et al. 2020; Zhu et al. 2020) and accuracy is used as an evaluation metric. More implementation details are in the supplementary materials.

4.3 Results on Node Classification

Table 2 shows the results of Deformable GCN and other baselines on node classification tasks. The best model for each dataset is highlighted with boldface. Overall, Deformable GCN achieves the highest performance on all heterophilic graphs compared to all baselines including H₂GCN, which is specifically proposed for heterophilic graphs. Note that on some heterophilic graph datasets such as Texas, Wisconsin, Actor, and Cornell, MLP outperforms various GNNs such as GCN, GAT, and Geom-GCN by significant margins without utilizing any graph structure information. It might mean that graph structure information is

formable on all the datasets. Similar to our method, datasets have the same task in node embedding. Deformable GCN is a node embedding-based method.

4.4 Ablation Study

We conduct ablation studies to verify the effectiveness of Deformable GCN.

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Ver Steeg, G.; and Galst'yan, A. 2019. Mishop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*.

Berg, R. v. d.; Kipf, T. N.; and Welling, M. 2017. Graph convolutional matrix completion. arXiv:1706.02263.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In *AAAI*.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable Convolutional Networks. In *ACCV*.

Deffernard, M.; Breuss, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*.

Erkan, G.; and Radev, D. R. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR*, 22: 457–479.

Erkic, F.; Podda, M.; Baccia, D.; and Micheli, A. 2019. A fair comparison of graph neural networks for graph classification. In *ICLR*.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *ICML*.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *KDD*.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2018. Inductive Representation Learning on Large Graphs. In *NeurIPS*.

Hammond, D. K.; Vandergheynst, P.; and Gibouval, R. 2011. Wavelets on graphs via spectral graph theory. *ACMA*, 30(2): 129–150.

He, K.; Girshick, R.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Johnson, J.; Krishna, R.; Stark, M.; Li, L.-J.; Sherris, D.; Bernstein, M.; and Fei-Fei, L. 2015. Image retrieval using scene graphs. In *CVPR*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Li, P.; Wang, Y.; Wang, H.; and Leskovec, J. 2020. Distance Encoding—Designing Powerful GNNs for Structural Representation Learning. In *NeurIPS*.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *ICML*.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-gcn: Geometric graph convolutional networks. In *ICLR*.

Rosenberczki, B.; Allen, C.; and Sarkar, R. 2021. Multi-scale attributed node embedding. *J. Complex Networks*, 9(2).

Schlichtkrull, M.; Kipf, T. N.; Bloom, P.; Van Den Berg, R.; Tsvy, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *ESWC*, 593–607.

Sen, P.; Namata, G.; Bilgic, M.; Gionis, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Siasolin, A.; Sanghavi, S.; Lathuillière, S.; and Sebe, N. 2018. Deformable GANs for Pose-Based Human Image Generation. In *CVPR*.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *WWW*.

Tang, J.; Sun, J.; Wang, C.; and Yang, Z. 2009. Social influence analysis in large-scale networks. In *SIGKDD*.

Thomas, H.; Qi, C. R.; Dechoud, F.-E.; Marconig, B.; Goulette, F.; and Guibas, L. J. 2019. Kgcnn: Flexible and deformable convolution for point clouds. In *ICCV*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. In *ICLR*.

Wang, D.; Cui, P.; and Zhu, W. 2016. Structural deep network embedding. In *KDD*.

Wang, X.; Chan, K. C. K.; Yu, K.; Dong, C.; and Loy, C. C. 2019. EDVR: Video Restoration With Enhanced Deformable Convolutional Networks. In *CVPR*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *ICML*.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation learning on graphs with jumping knowledge networks. In *ICML*.

Ying, R.; You, J.; Morris, C.; Ren, X.; Hamilton, W. L.; and Leskovec, J. 2018. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*.

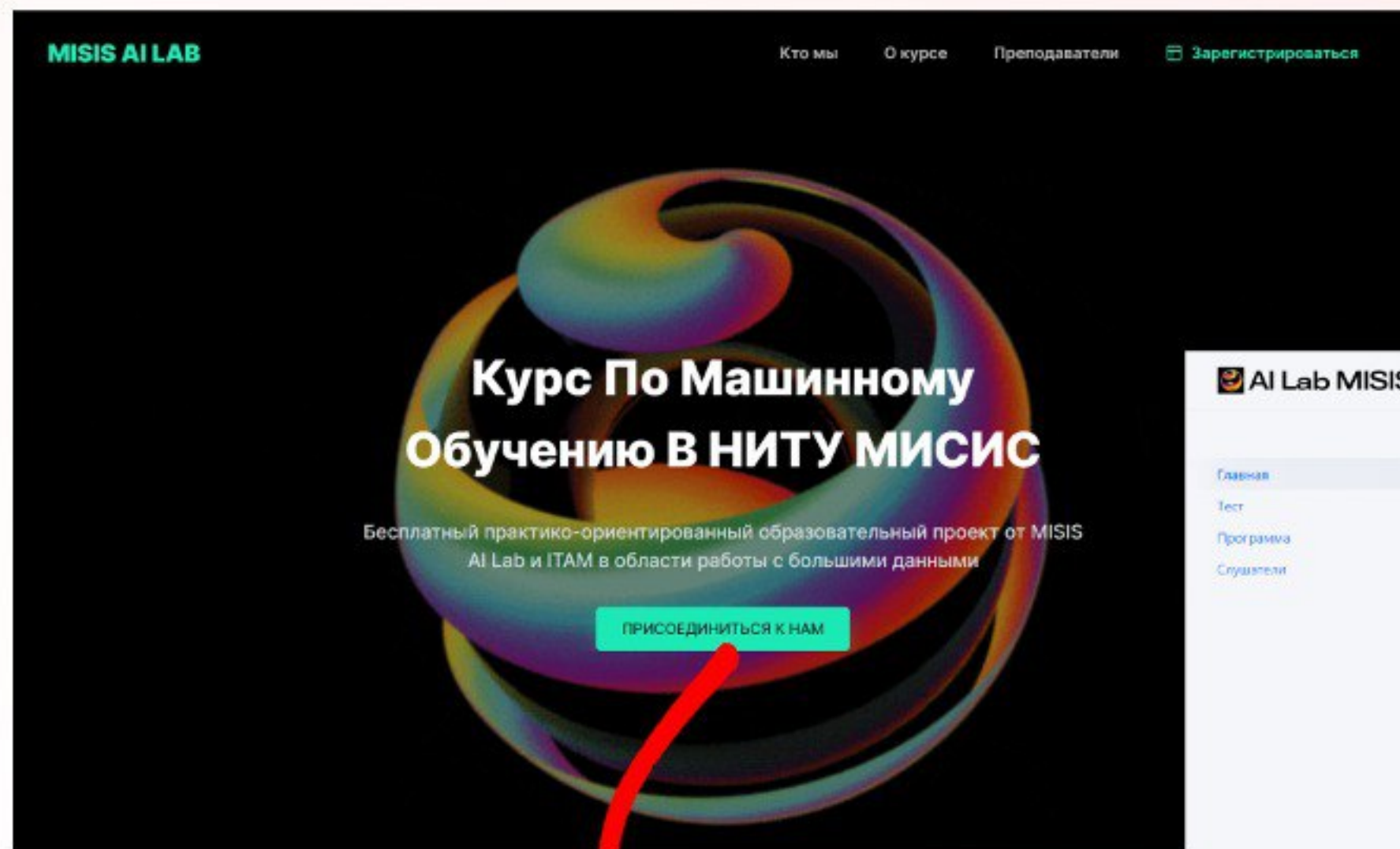
Yun, S.; Jeong, M.; Kim, R.; Kang, J.; and Kim, H. J. 2019. Graph Transformer Networks. In *NeurIPS*.

Zhang, M.; and Chen, Y. 2018. Link prediction based on graph neural networks. In *NeurIPS*.

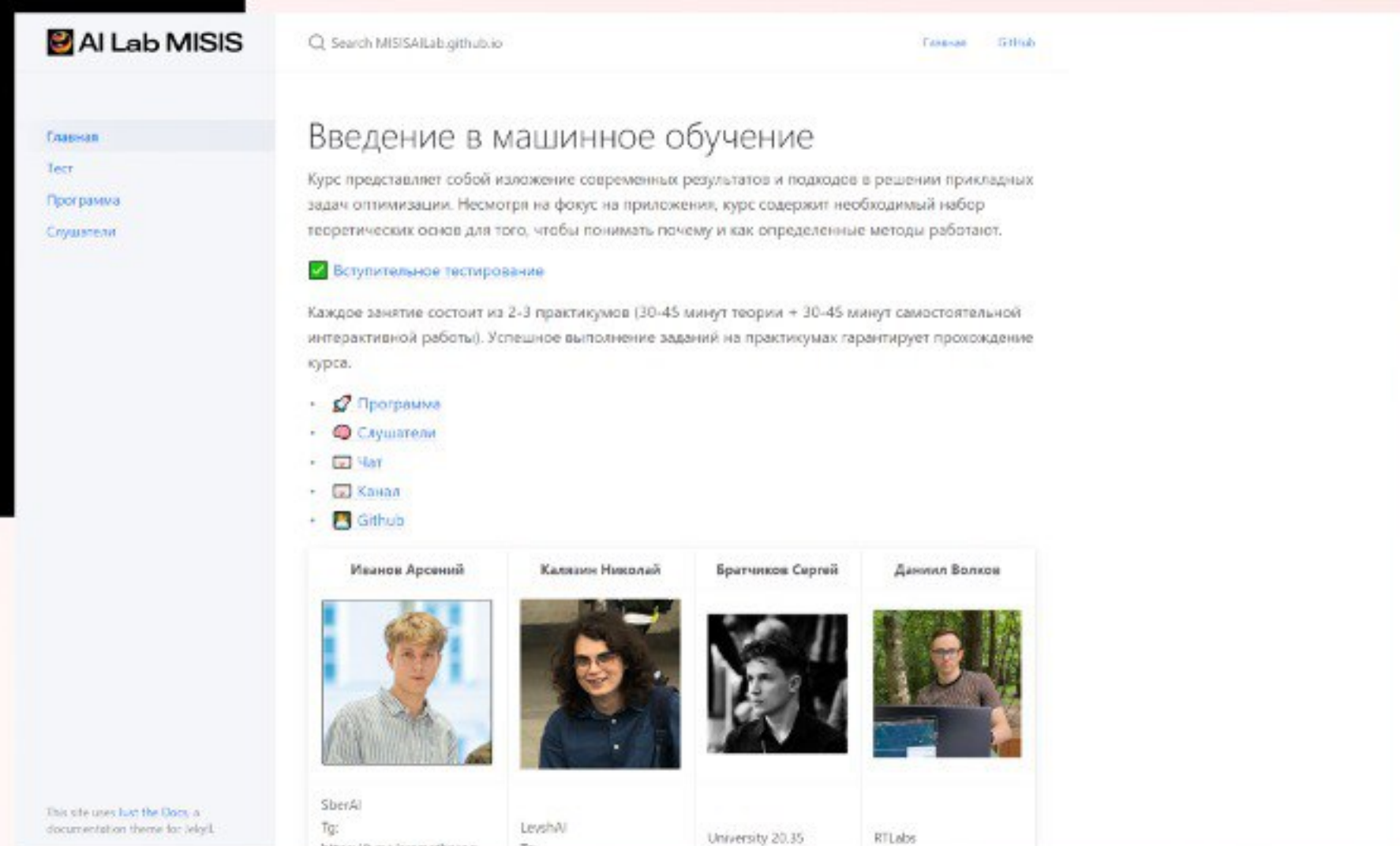
Zhu, J.; Rossi, R. A.; Rao, A.; Mai, T.; Lipka, N.; Ahmed, N. K.; and Koutra, D. 2021. Graph Neural Networks with

Machine Learning course

AI Lab MISIS



Все в одном месте!





Николай Калязин

- 3 курс Прикладная математика
- ex сотрудник SberDevices
- работаю в стартапе LevshAI

Изучаю и работаю в CV, NLP

Libs: Pandas, NumPy, SciPy, PyTorch, TensorFlow



Machine Learning – is....

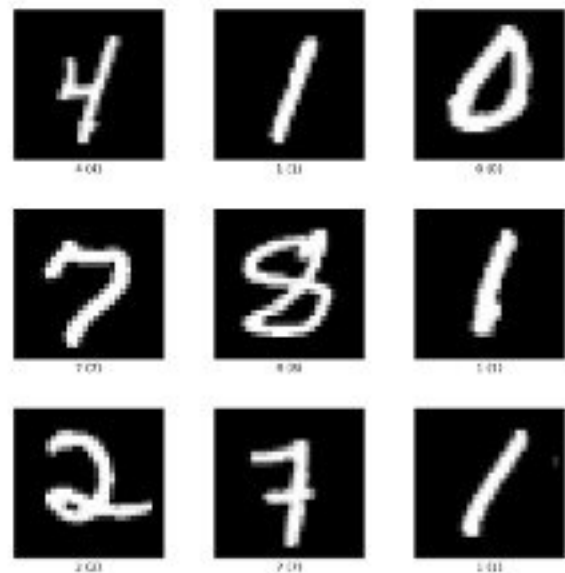
1. **“Машинное обучение – это наука, изучающая алгоритмы, которые способны извлекать закономерности из ограниченного количества примеров и улучшаться за счет опыта”**
2. **“Алгоритм машинного обучения – это такой алгоритм, который способен обучаться на данных.”**

Говорят, что программа обучается на опыте E относительно некоторого класса задач T и меры качества P , если качество на задачах из T , измеренное с помощью P , возрастает с ростом опыта E .



Tom Michael Mitchell

Machine learning



Задача: Распознавание символов

Мера: Процент правильно распознанных

Опыт: База, размеченных вручную, символов

Задача: Игра в шашки, шахматы, го

Мера: Процент побед

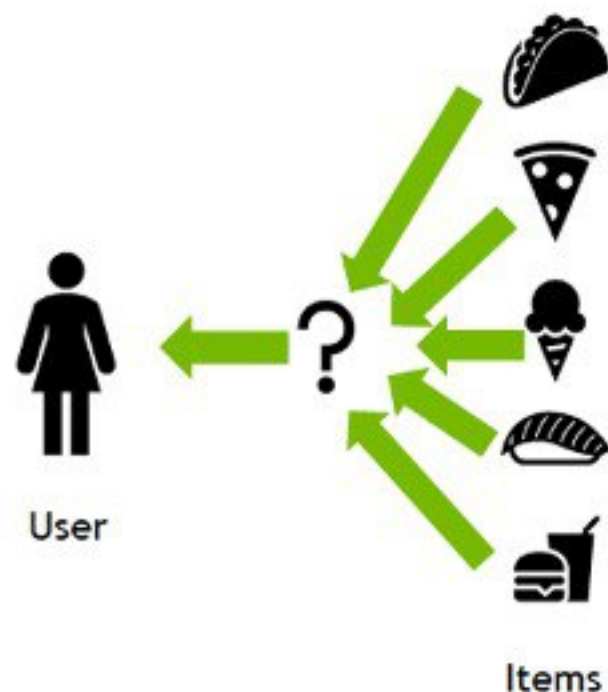
Опыт: Игра программы против себя



Задача: Рекомендация товаров, услуг, видео

Мера: Процент успешных рекомендаций

Опыт: Список товаров, просмотренных, купленных, оцененных пользователями



ML THESAURUS

Name

Age

Statistics

Python

Eye color

Native language

Target (mark)

Target (passed)

ML thesaurus

Observation (or datum, or data point) is one piece of information.

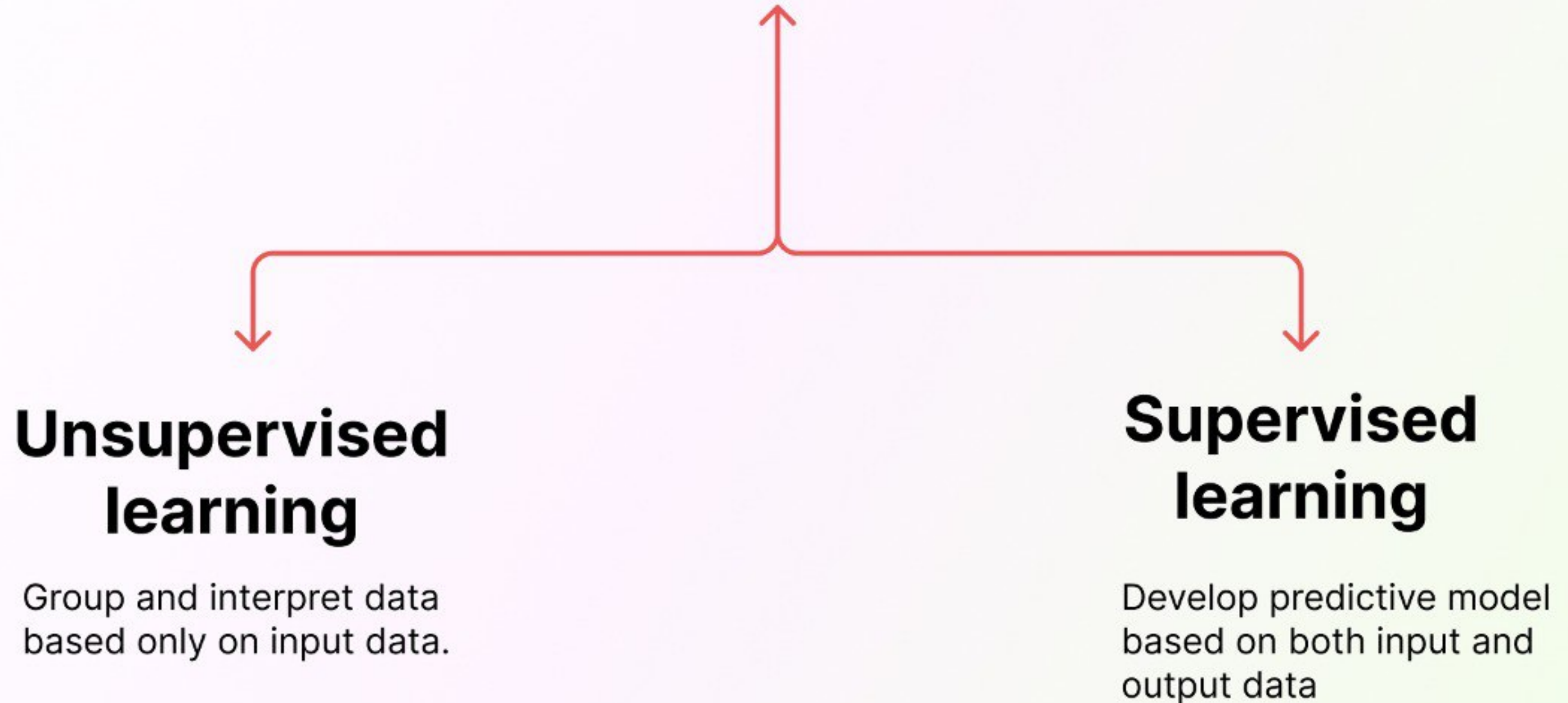
Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Target (passed)
John	22	5	4	Brown	English	5	TRUE
Aahna	17	4	5	Brown	Hindi	4	TRUE
Emily	25	5	5	Blue	Chinese	5	TRUE
Michael	27	3	4	Green	French	5	TRUE
Some student	23	3	3	NA	Esperanto	2	FALSE

In many cases the observations are supposed to be *i.i.d.*

- *independent*
- *identically distributed*

Dataset

Machine Learning Tasks

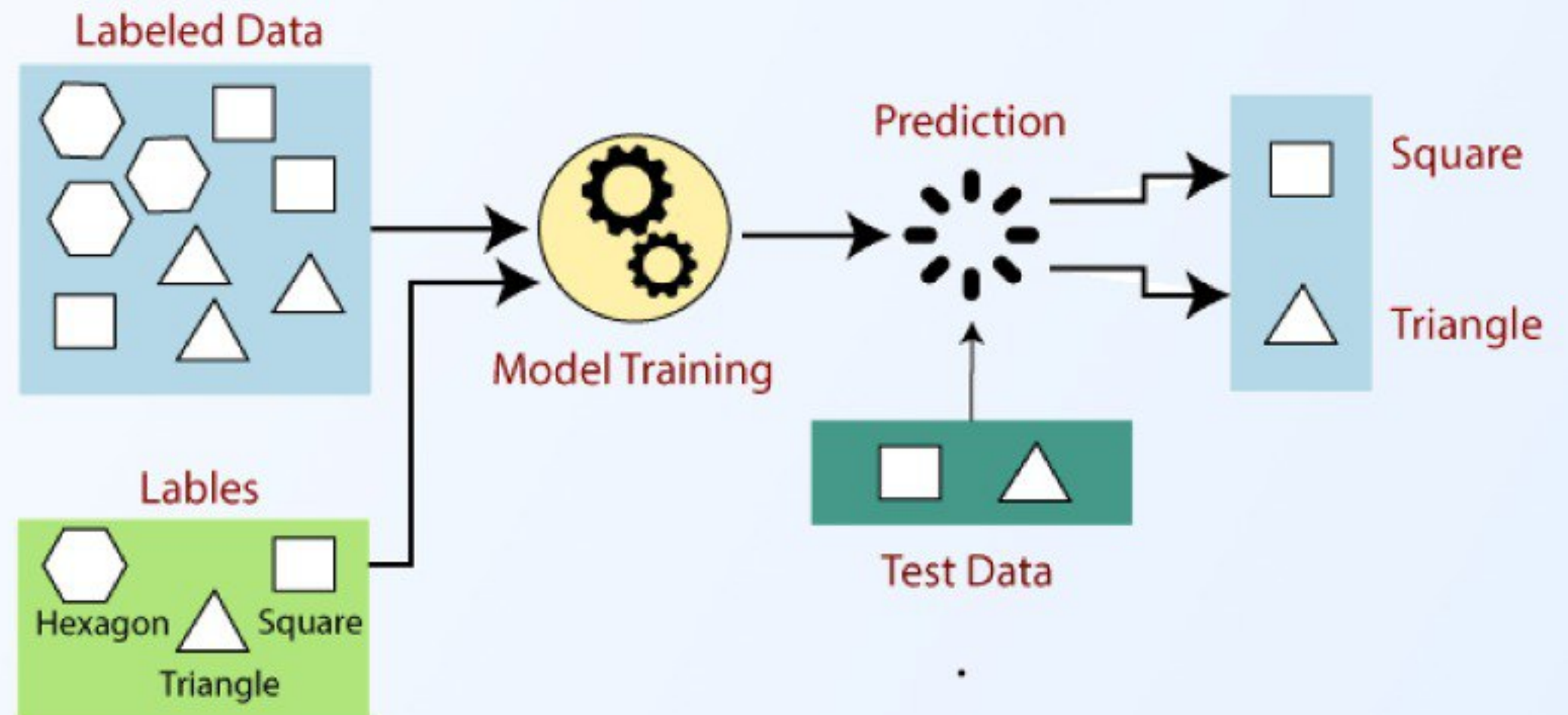


Supervised learning

Добавим чуточку математических обозначений:

Пусть у нас есть датасет $X_{train} = \{(x_1, y_1), \dots, (x_n, y_n)\}$. Где x_i - объекты, y_i - таргеты. Как уже было сказано, мы хотим построить отображение $f: X \rightarrow Y$, где X - пространство объектов, а Y - пространство таргетов.

У нас есть объекты, мы хотим восстановить целевую зависимость, чтобы уметь выдавать ответы и на новые объекты тоже.

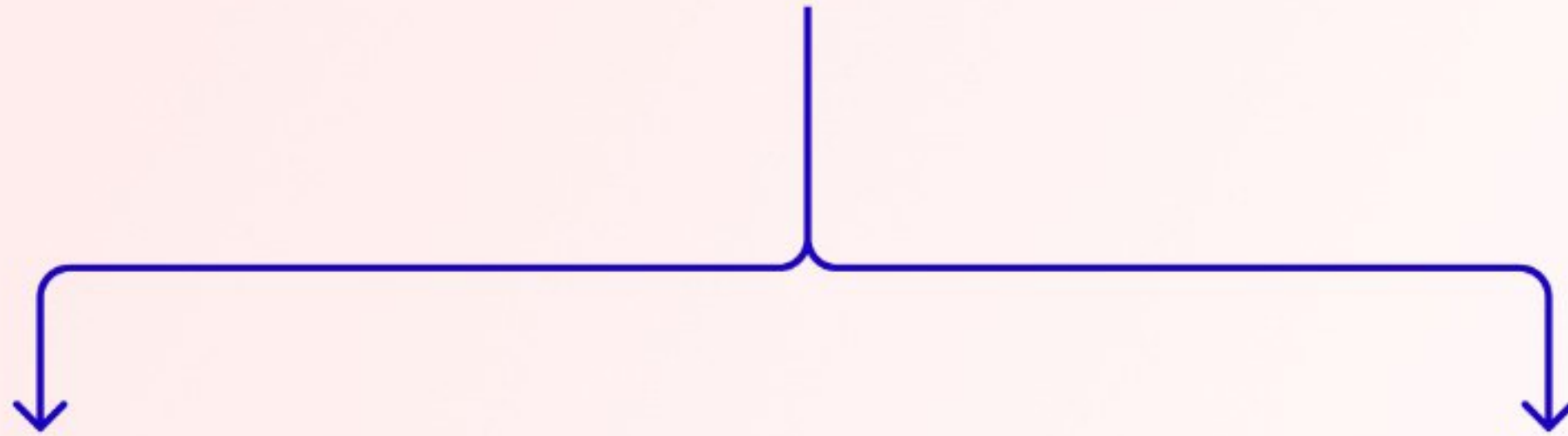


Supervised target learning tasks

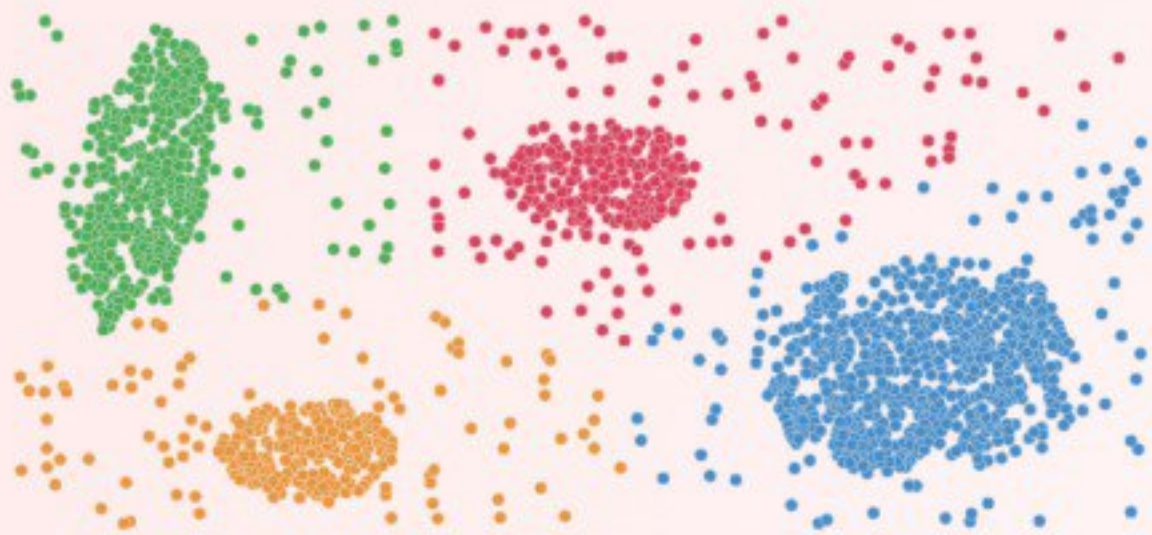
Между собой задачи в обучении с учителем отличаются пространством таргетов.



Unsupervised learning



Кластеризация



Понижение размерности

