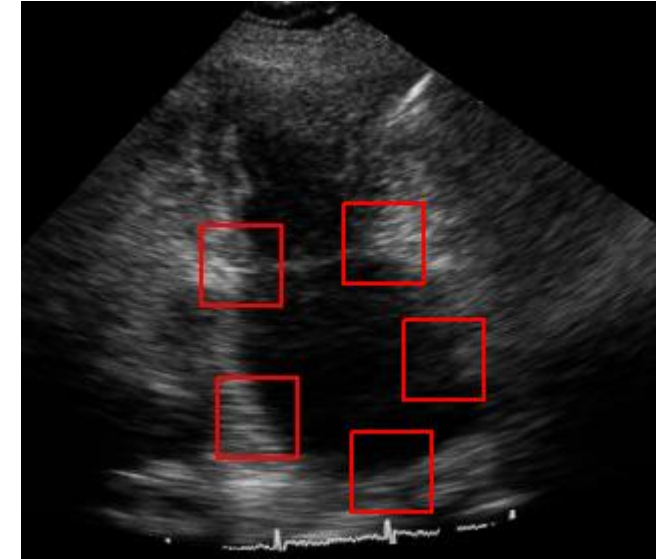
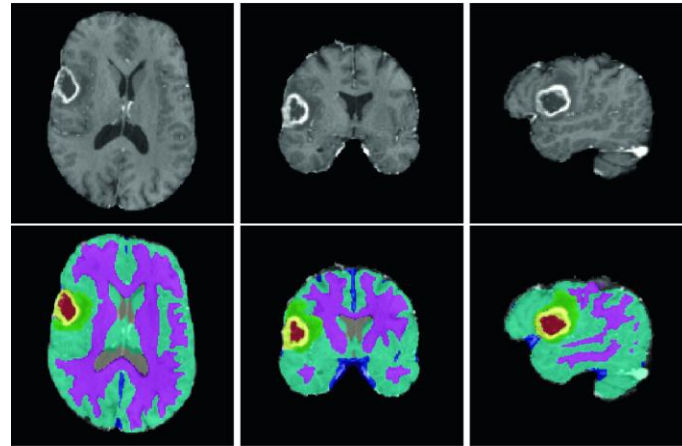


# Лабораторная работа 2. Использование открытых баз данных для распознавания медицинских изображений



Цели работы:

1. Ознакомиться с площадкой для проведения соревнований по машинному обучению Kaggle.
2. Ознакомиться с инструментами языка Python 3 для задач машинного обучения.



# Что такое Kaggle?

## Соревнования

### All Competitions

Active (Not Entered) Completed InClass

All Categories ▾ Default Sort ▾



#### HuBMAP - Hacking the Kidney

Identify glomeruli in human kidney tissue images

Research • a month to go • Code Competition • 1087 Teams

\$60,000



#### RANZCR CLIP - Catheter and Line Position Challenge

Classify the presence and correct placement of tubes on chest x-rays to save lives

Featured • 20 days to go • Code Competition • 1066 Teams

\$50,000



#### Human Protein Atlas - Single Cell Classification

Find individual human cell differences in microscope images

Featured • 3 months to go • Code Competition • 202 Teams

\$25,000



#### Indoor Location & Navigation

Identify the position of a smartphone in a shopping mall

Research • 3 months to go • 349 Teams

\$10,000

## Курсы



### Python

Learn the most important language for data science.



### Intro to Machine Learning

Learn the core ideas in machine learning, and build your first models.



### Intermediate Machine Learning

Learn to handle missing values, non-numeric values, data leakage and more. Your models will be more accurate and useful.



### Data Visualization

Make great data visualizations. A great way to see the power of coding!

## Способ заявить о себе

Competitions Datasets Notebooks Discussion [Learn more about rankings >](#)



205  
Grandmasters



1,546  
Masters



6,414  
Experts



58,523  
Contributors



89,044  
Novices

Rank	Tier	User	Medals	Points
1		Psi	joined 9 years ago  16  6  0	220,282
2		Guanshuo Xu	joined 5 years ago  17  16  2	216,347
3		Dieter	joined 3 years ago  16  8  3	196,374
4		bestfitting	joined 4 years ago  29  9  1	191,335
5		Μarioς Μιχαηλίδης KazAnova	joined 8 years ago  39  54  38	144,812
6		Giba	joined 9 years ago  55  44  28	130,909

Почему стоит этим заняться:

- Получение опыта
- Можно использовать в резюме
- Большое сообщество людей у которых есть чему научиться

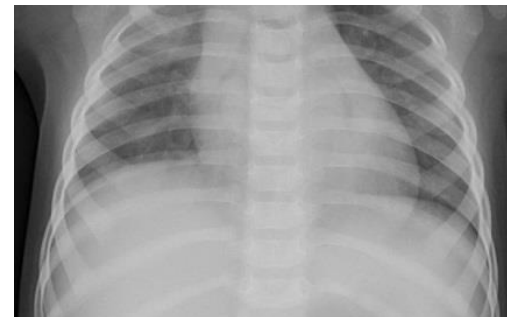
# Датасет

Набор рентгеновских снимков  
грудной клетки 2 классов:

Норма



Пневмония



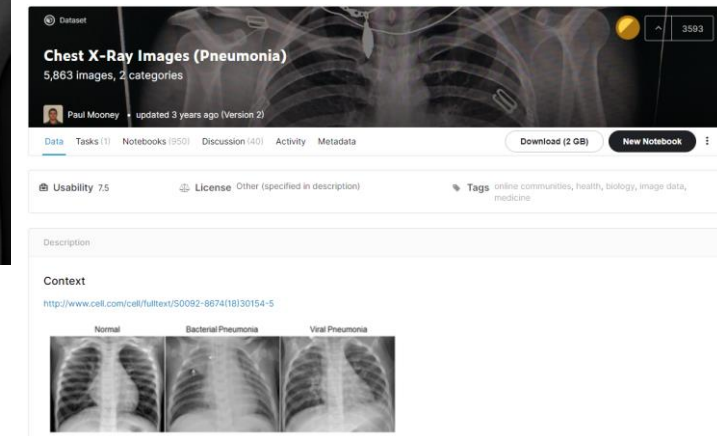
Снимки распределены по трем  
папкам:

Train: 1341 снимков – норма, 3875  
– пневмония.

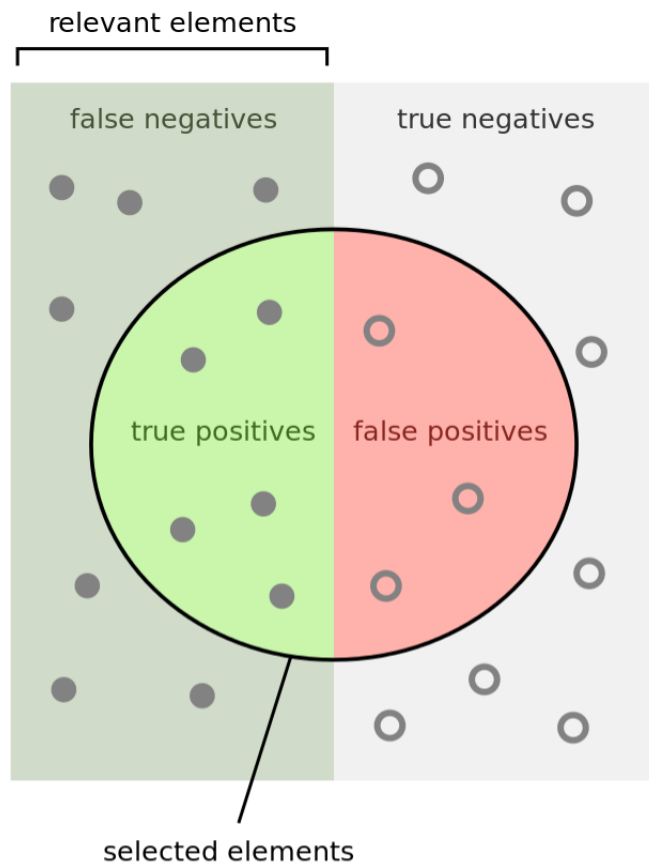
Val: 8 снимков – норма, 8 –  
пневмония.

Test: 234 снимков – норма, 390 –  
пневмония.

- Все рентгеновские снимки грудной клетки выполнялись как часть обычного клинического ухода за пациентами
- Низкокачественные и нечитаемые снимки были исключены
- Диагнозы были поставлены двумя опытными врачами
- Для исключения ошибок полученный результаты были проверены третьим экспертом



# Метрика для несбалансированных наборов данных



$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

**Precision:** Какая часть положительных предсказаний была на самом деле правильной?

**Recall:** Какая доля реальных положительных примеров была определена правильно?

**F1:** Использует Precision и Recall для общей оценки.

# Где работать с данными, если нет мощного компьютера?

## Kaggle Notebooks

### Code

Explore and run machine learning code with Kaggle Notebooks. Find help in the [Documentation](#).

+ New Notebook

Your work

🔍 Search public notebooks

Python

R

Beginner

NLP

Finance

Random Forest

GPU

TPU

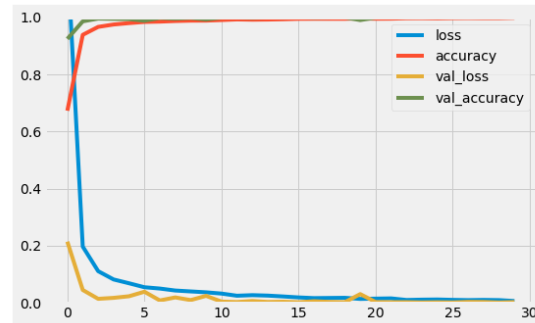
Competition notebook

## Что можно делать?

- Писать код
- Управлять версиями
- Публиковать код
- Обсуждать свои и чужие решения
- Изучать код других людей

### Evaluating the model

```
In [14]: pd.DataFrame(history.history).plot(figsize=(8, 5))
plt.grid(True)
plt.gca().set_ylim(0, 1)
plt.show()
```



### Splitting the data into train and validation set

```
In [9]: X_train, X_val, y_train, y_val = train_test_split(image_data, image_labels, test_size=0.3, random_state=42, shuffle=True)

X_train = X_train/255
X_val = X_val/255

print("X_train.shape", X_train.shape)
print("X_val.shape", X_val.shape)
print("y_train.shape", y_train.shape)
print("y_val.shape", y_val.shape)

X_train.shape (27446, 30, 30, 3)
X_val.shape (11763, 30, 30, 3)
y_train.shape (27446,)
y_val.shape (11763,)
```

### Competition Notebooks

See all (125)

**Ranzcr 🤖 ResNet200D + SeResNet152D inference**  
Updated 17 hours ago  
pytorch\_images\_seresnet+3

14

**mmdetection 0.260+ baseline**  
Updated a day ago  
VinBigData Chest X-ray Abnormalities Detection

25

**[日本語]pytorch\_starter**  
Updated a day ago  
pytorch image models+1

17

**RANZCR CLiP - GroupKFold with TFRecords**  
Updated a day ago  
RANZCR CLiP - Catheter and Line Position Challenge+1

12

# Язык программирования



```
CLASS torch.utils.data.DataLoader(dataset: torch.utils.data.dataset.Dataset[T_co],
    batch_size: Optional[int] = 1, shuffle: bool = False, sampler:
    Optional[torch.utils.data.sampler.Sampler[int]] = None, batch_sampler:
    Optional[torch.utils.data.sampler.Sampler[Sequence[int]]] = None, num_workers:
    int = 0, collate_fn: Callable[List[T], Any] = None, pin_memory: bool = False,
    drop_last: bool = False, timeout: float = 0, worker_init_fn: Callable[int, None]
    = None, multiprocessing_context=None, generator=None, *, prefetch_factor: int =
    2, persistent_workers: bool = False)
```

Data loader. Combines a dataset and a sampler, and provides an iterable over the given dataset.

The `Dataloader` supports both map-style and iterable-style datasets with single- or multi-process loading, customizing loading order and optional automatic batching (collation) and memory pinning.

See [torch.utils.data](#) documentation page for more details.

## Parameters

- **dataset** (*Dataset*) – dataset from which to load the data.
- **batch\_size** (*int, optional*) – how many samples per batch to load (default: 1).
- **shuffle** (*bool, optional*) – set to `True` to have the data reshuffled at every epoch (default: `False`).
- **sampler** (*Sampler or Iterable, optional*) – defines the strategy to draw samples from the dataset. Can be any



## Почему Python?

- Быстро осваивается
- Больше количество руководств и документации
- Интерактивность
- Множество библиотек и фреймворков для машинного обучения

```
1 # Python
2 # Use Decimal to do high precision calculation
3
4 >>> 2.36 * 5.1
5 12.035999999999998
6
7 >>> from decimal import Decimal
8 >>> result = Decimal("2.36") * Decimal("5.1")
9 >>> result
10 Decimal('12.036')
11 >>> a = float(result)
12 >>> a
13 12.036
14
```





# Фреймворк машинного обучения



```
a = torch.tensor([1,2,3,4], dtype = torch.float64)
print(a)
```

```
tensor([1., 2., 3., 4.], dtype=torch.float64)
```

```
a = torch.randn((3,3,3))
print(a)
```

```
tensor([[[ 0.8834,  3.1001,  0.1028],
         [-0.7967, -0.8711, -1.3769],
         [-1.6110,  0.5500, -1.6429]],

        [[-1.3078, -0.2058,  0.4480],
         [-0.2859, -1.5177,  0.7204],
         [ 1.3631,  0.5193, -0.6224]],

        [[ 0.6159, -1.8170,  0.3920],
         [ 1.0218, -0.6155, -0.6653],
         [-1.9467, -0.4836, -0.4640]]])
```

```
net = torch.nn.Sequential(
    torch.nn.Linear(3, 4),
    torch.nn.Sigmoid(),
    torch.nn.Linear(4, 1),
    torch.nn.Sigmoid()
)
print(net)
```

```
Sequential(
  (0): Linear(in_features=3, out_features=4, bias=True)
  (1): Sigmoid()
  (2): Linear(in_features=4, out_features=1, bias=True)
  (3): Sigmoid()
)
```

```
a = torch.randn((2,4), dtype=torch.float32, requires_grad=True)
b = torch.randn((4,1), dtype=torch.float32, requires_grad=True)
c = a.mm(b)
d = c.sum()

print(a)
print(b)
print(c)
print(d)
```

```
tensor([[ 1.3776,  0.3604, -0.3725, -0.5138],
        [ 0.9580, -0.7806, -0.9501,  0.4071]], requires_grad=True)
tensor([[ -0.7947],
        [ 1.2969],
        [ 0.1652],
        [-0.7777]], requires_grad=True)
tensor([[-0.2894],
        [-2.2473]], grad_fn=<MmBackward>)
tensor(-2.5367, grad_fn=<SumBackward0>)
```

```
d.backward()
print(a.grad)
print(b.grad)
```

```
tensor([[ -0.7947,  1.2969,  0.1652, -0.7777],
        [-0.7947,  1.2969,  0.1652, -0.7777]])
tensor([[- 2.3356],
        [-0.4202],
        [-1.3226],
        [-0.1067]])
```

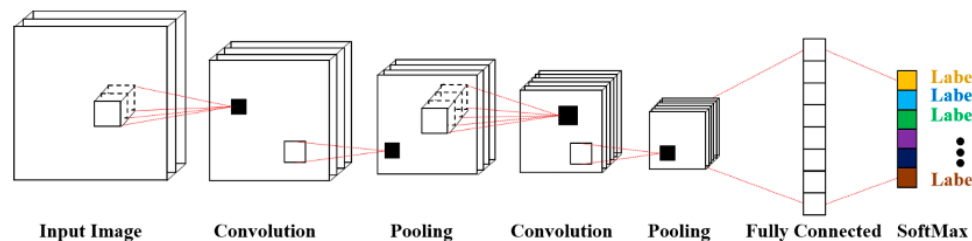
```
if torch.cuda.is_available():
    model = model.to('cuda')
```

Что можно делать?

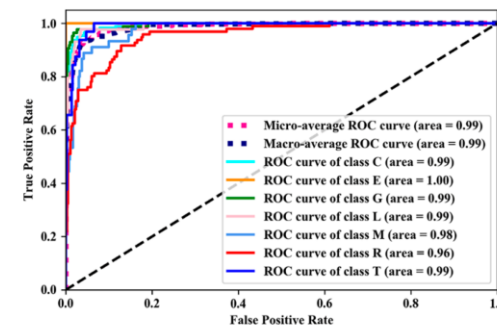
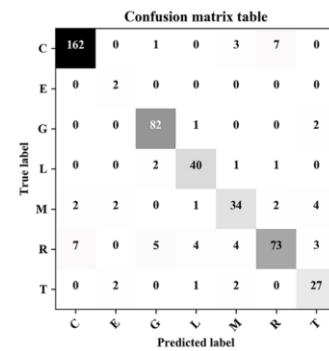
- Работать с тензорами
- Автоматически вычислять градиент
- Использовать конструктор для построения своих моделей
- Все это на GPU

# Что нужно сделать?

Создать и обучить классификатор  
рентгеновских снимков для  
диагностирования пневмонии.



Произвести оценку модели



Представить свое решение с  
кодом для обучения итоговой  
модели :

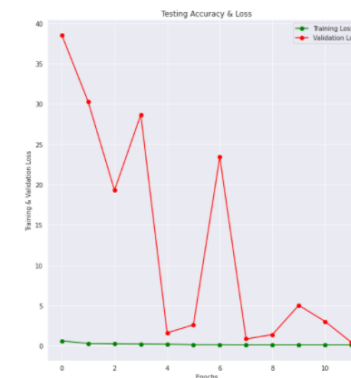
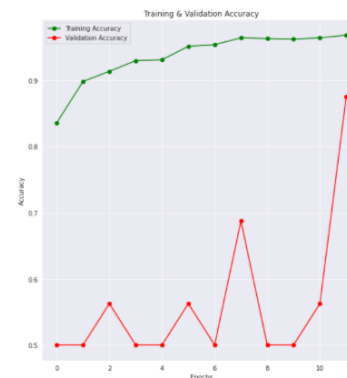
- Обосновать выбор архитектуры
- Описать особенности процесса обучения и подготовки данных
- Привести данные об оценке полученной модели
- Опубликовать свое решение на Kaggle\*.



## Pneumonia Detection using CNN(92.6% Accuracy)

Python notebook using data from [Chest X-Ray Images \(Pneumonia\)](#) · 25,027 views · 8mo ago

```
ax[1].set_title('Testing Accuracy & Loss')  
ax[1].legend()  
ax[1].set_xlabel("Epochs")  
ax[1].set_ylabel("Training & Validation Loss")  
plt.show()
```



```
In [122]:  
predictions = model.predict_classes(x_test)  
predictions = predictions.reshape(1,-1)[0]  
predictions[:15]
```

174 Copy and Edit 426

Version 3 of 3

Quick Version

Notebook

What Is Pneumonia?

Importing The  
Necessary Libraries

Description Of The  
Pneumonia Dataset

Loading The Dataset

Data Visualization &  
Preprocessing

Data Augmentation

Training The Model

Analysis After Model  
Training

Input (1)

Execution Info

Log

Comments (72)

[Pneumonia Detection using CNN\(92.6% Accuracy\) | Kaggle](#)