```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-pyt
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all fil

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preser
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside
```

⍰Pnevmoniya haqida qisqacha:

Xalqaro miqyosda 5 yoshgacha bo'lgan bolalar o'limining 15% dan ortig'ini pnevmoniya tashkil qiladi. 2015-yilda 5 yoshgacha bo'lgan 920 ming bola ushbu kasallikdan vafot etgan. Amerika Qo'shma Shtatlarda pnevmoniya tez yordam bo'limi yiliga 500 000 dan ortiq tashriflar va 2015 yilda 50 000 dan ortiq o'limni tashkil qilgan, bu kasallik mamlakatda o'limning eng yaxshi 10 sabablari ro'yxatida saqlanadi.

Pnevmoniya keng tarqalgan bo'lsa-da, aniq tashxis qo'yish juda qiyin. Bu yuqori malakali mutaxassislar tomonidan ko'krak qafasi rentgenogrammasini (CXR) ko'rib chiqishni va klinik tarix, hayotiy belgilar va laboratoriya tekshiruvlari orqali tasdiqlashni talab qiladi. Pnevmoniya odatda CXRda xiralashgan joy yoki xiralik ko'rinishida namoyon bo'ladi. Biroq, CXRda pnevmoniya diagnostikasi o'pkada suyuqlikning ortiqcha yuklanishi (o'pka shishi), qon ketishi, hajmning yo'qolishi (atelektaziya yoki kollaps), o'pka saratoni yoki radiatsiya yoki jarrohlikdan keyingi o'zgarishlar kabi bir qator boshqa holatlar tufayli murakkablashadi. O'pkadan tashqarida plevra bo'shlig'idagi suyuqlik (plevral efüzyon) ham CXRda shaffoflikning oshishi sifatida namoyon bo'ladi. Mavjud bo'lsa, bemorning turli vaqt nuqtalarida olingan CXRlarini taqqoslash va klinik simptomlar va tarix bilan bog'liqligi tashxis qo'yishda yordam beradi.

Biz ushbu amaliyotda insonlarning ko'krak qafasi rentgent suratiga ko'ra ularning pnevmoniyaga chalingan yoki yo'qligini aniqlaymiz !

⍰Kerakli kutubxonalar⍰✔:

```
import torch
import fastai
import seaborn as sns
from fastai.vision.all import *
from tqdm import tqdm
from torch.utils.data import dataloader
```

```
print(torch.__version__)
print(fastai.__version__)
```

⍰Joylashuv:

```
path = Path('../input/pnevmoniya/train')
fls = get_image_files(path)
```

tarkib:

```
os.listdir('../input/pnevmoniya/train') # tarkib
```

Bizning ma'lumotlarimiz ikkita jildda joylashgan:

1. train= bizning modelimizni o'rgatish uchun o'quv ma'lumotlarini/tasvirlarini o'z ichiga oladi.
2. test = bu tasvirlar va ularning yorlig'i (pnevmoniya/pnevmoniya emas) o'rtasidagi munosabatlarni o'rgangandan so'ng modelni sinab ko'rish uchun foydalanadigan ma'lumotlarni o'z ichiga oladi.

```
main_dir = os.listdir('../input/pnevmoniya/train') # asosiy data
print(main_dir)
```

```
# train
train_folder = "../input/pnevmoniya/train/"
# test
test_folder = '../input/pnevmoniya/test/'
```

⍰Keling, ba'zi rasmlarni ko'rib chiqaylik.

```
# train dan normal va pnevm rasmlar ni ajratib olish
os.listdir(train_folder)
# norm
train_n = train_folder+'NORMAL/'
# pnev
train_p = train_folder+'PNEUMONIA/'
```

```
#Normal pictures

# train data dagi normal rasmlar soni:
print('train_n soni: ',len(os.listdir(train_n)))
# 0 dan len(os.listdir(train_n)) gacha tasodifiy butun son:
rand_norm = np.random.randint(0,len(os.listdir(train_n)))
# shu rand_norm ga mos keluvchi rasm:
norm_pic = os.listdir(train_n)[rand_norm]
# rasmning nomi
```

```
print('normal rasm nomi: ', norm_pic)
# normal rasmning manzili:
norm_pic_address = train_n + norm_pic







#Pnevmoniya mavjud bo`lgan rasm:


# tasodifiy index:
print('train_p soni: ',len(os.listdir(train_p)))
rand_p = np.random.randint(0,len(os.listdir(train_p)))
# image fayl nomi:
pnev_pic =  os.listdir(train_p)[rand_norm]
# pnev_pic ning toliq manzili:
pnev_address = train_p + pnev_pic
print('pnevmo rasm nomi:', pnev_pic)

# Rasmlarni o`qish:
norm_load = Image.open(norm_pic_address)
pnev_load = Image.open(pnev_address)

# Rasmlarni grafik ravishda chop etish:
f = plt.figure(figsize= (10,6))
a1 = f.add_subplot(1,2,1)
img_plot = plt.imshow(norm_load)
a1.set_title('Normal')

a2 = f.add_subplot(1, 2, 2)
img_plot = plt.imshow(pnev_load)
a2.set_title('Pneumonia')
```

⍰⍰DataBlock yaratamiz:

```
# DataBlock yaratamiz:
pnevmoniya = DataBlock(
    blocks = (ImageBlock, CategoryBlock),
    get_items = get_image_files,
    splitter = RandomSplitter(valid_pct = 0.1, seed = 7),
    get_y = parent_label,
    item_tfms = Resize(128)
)


# dataloaders:
dls = pnevmoniya.dataloaders(path)

# Train qilamiz:
```

```
learn = cnn_learner(dls, resnet50, metrics = accuracy)
learn.fine_tune(1) #  1 marta fine_tune qiladi(tekshirib chiqadi)
```

⍰Modelimiz 96% aniqlik bilan ishlamoqda⍰⍰

⍰confusion matrix (visual ko'rinishga keltirish)

```
interp = ClassificationInterpretation.from_learner(learn)
interp.plot_confusion_matrix()
```

```
test_file = !ls "../input/pnevmoniya/test"

pred_id = []
predicted = []

for file in tqdm(test_file):
    pred, n_th, probs = learn.predict(f"../input/pnevmoniya/test/{file}")
    pred_id.append(n_th)
    predicted.append(pred)
```

```
bashorat = []
for num in predicted:
    if num == 'NORMAL':
        bashorat.append(0)
    elif num == 'PNEUMONIA':
        bashorat.append(1)
```

```
data = pd.read_csv('../input/pnevmoniya/sample_solution.csv')
data.head()
```

```
data['labels'] = bashorat
data.head()
```

Bashorat qilingan datani saqlaymiz

```
data.to_csv('pnevmoniya_01.csv', index=False)
```

⍰⍰E'tiboringiz uchun Tashakkur⍰