**Термины и сокращения**

PyLib - путь до папки в домашней директории, где будет лежать или уже лежит Python, например, /home/ngusarova/Kaggle5/py-365

BzLib - путь до папки в домашней директории, где будет лежать или уже лежит bzip2, например, /home/ngusarova/Kaggle5/bzip2

BazelLib - путь до папки в домашней директории, где будет лежать или уже лежит Bazel, например, /home/ngusarova/Kaggle5/bazel

**Настройка окружения**

bzip2

wget <http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz>

tar xf bzip2-1.0.6.tar.gz

cd bzip2-1.0.6

make

mkdir BzLib

make install PREFIX=BzLib

Python3

wget <https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tgz>

tar xf Python-3.6.5.tgz

cd Python-3.6.5

mkdir PyLib

./configure --prefix=PyLib

nano setup.py

Находим раздел, где подключается модуль bz2, там должно быть условие self.compiler.find\_library\_file, в котором мы явно указываем, где искать необходимое расширение: if (self.compiler.find\_library\_file(['BzLib/lib'], 'bz2'))

cd Include

ln -s BzLib/include/bzlib.h bzlib.h

cd ..

make

**Убедитесь, что в выводе в конце make среди «The necessary bits to build these optional modules were not found:» нет \_bz2, или среди «Failed to build these modules» нет \_bz2**

make install

Для упрощения работы можно сделать:

alias python3=PyLib/bin/python3

alias pip3=PyLib/bin/pip3

Или добавить ссылки в директории, которые перечислены в PATH, в случае Ломоносов-2:

ln -s PyLib/bin/python3 ~/.local/bin/python3

ln -s PyLib/bin/pip3 ~/.local/bin/pip3

JDK

wget --no-check-certificate -c --header "Cookie: oraclelicense=accept-securebackup-cookie" <http://download.oracle.com/otn-pub/java/jdk/8u161-b12/2f38c3b165be4555a1fa6e98c45e0808/jdk-8u161-linux-x64.tar.gz>

Если ссылка не верна, то заходим [сюда](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html), принимаем соглашение и копируем адрес ссылки с jdk-8u161-linux-x64.tar.gz

tar xf jdk-8u161-linux-x64.tar.gz

ln -s <Папка, где распакован jdk>/bin/javac ~/.local/bin/javac

Bazel

wget <https://github.com/bazelbuild/bazel/releases/download/0.4.2/bazel-0.4.2-installer-linux-x86_64.sh>

chmod +x bazel-0.4.2-installer-linux-x86\_64.sh

./bazel-0.4.2-installer-linux-x86\_64.sh --prefix=BazelLib --bazelrc=BazelLib/.bazelrc

ln -s BazelLib/bin/bazel ~/.local/bin/bazel

Tensorflow

wget <https://github.com/pfillard/tensorflow/archive/r1.0_relu1.zip>

unzip tensorflow-r1.0\_relu1.zip

cd tensorflow-r1.0\_relu1

pip3 install numpy dev wheel mock

vim ./configure

Заменяем на 25 строке bazel clean --expunge на bazel clean --expunge\_async

./configure

Нажимаем enter на всех вопросах, кроме:

* + - Please specify the location of python

PyLib/bin/python3

* + - Please input the desired Python library path to use

PyLib/lib/python3.6/site-packages

bazel --batch --batch\_cpu\_scheduling build -c opt //tensorflow/tools/pip\_package:build\_pip\_package

bazel-bin/tensorflow/tools/pip\_package/build\_pip\_package /tmp/tensorflow\_pkg

pip3 install /tmp/tensorflow\_pkg/tensorflow-1.0.0rc0-py2-none-any.whl

skimage

pip3 install matplotlib numpy pil scipy

pip3 install scikit-build

pip3 install pillow --global-option="--disable-jpeg"

pip3 install scikit-image

Прочие зависимости

pip3 install xgboost SimpleITK h5py argparse

**Конвертирование данных**

wget <https://itk.org/ITKExamples/src/IO/GDCM/ReadDICOMSeriesAndWrite3DImage/ReadDICOMSeriesAndWrite3DImage.zip>

mkdir ReadDICOMSeriesAndWrite3DImage

unzip ReadDICOMSeriesAndWrite3DImage.zip -d ReadDICOMSeriesAndWrite3DImage

cd ReadDICOMSeriesAndWrite3DImage

python3 Code.py <Папка, где лежат DICOM>

Сконвертированные файлы будут лежать в той же папке, где были DICOM

**Тестирование**

wget <https://github.com/pfillard/tpx-kaggle-dsb2017/archive/master.zip>

unzip master.zip

cd tpx-kaggle-dsb2017-master

python3 -m predict -I <Путь до stage2\_sample\_submission.csv> -d <Путь до сконвертированного .nrrd-файла>

В консоль должно быть выведено:

saving predictions to predict\_results/predictions.csv

predicting done.