Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2.14
дисциплины «Программирование на Python»
Вариант №2

Выполнила:
Беседина Инга Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических
наук, доцент, доцент кафедры
инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: Установка пакетов в Python. Виртуальные окружения

Цель: приобретение навыков по работе с менеджером пакетов pip и виртуальными окружениями с помощью языка программирования Python версии 3.x

Ход работы

```
(Python_LR_2_14) C:\Users\Admин\Python_LR_2_14>
```

Рисунок 1. Виртуальное окружение Anaconda

```
(Python_LR_2_14) C:\Users\Admин\Python_LR_2_14>conda list
# packages in environment at C:\ProgramData\anaconda3\envs\Python_LR_2_14:
#
# Name                    Version                   Build  Channel
asgiref                   3.2.3                     py_0
blas                      1.0                         mkl
ca-certificates           2023.12.12            haa95532_0
certifi                   2020.6.20           pyhd3eb1b0_3
django                    3.2                 pyhd3eb1b0_0
icc_rt                    2022.1.0              h6049295_2
intel-openmp              2022.0.0            haa95532_3663
krb5                      1.16.1                h467bc42_7
libpq                     11.2                  h4410098_0
mkl                       2020.2                      256
numpy                     1.14.2             py27h0bb1d87_0
openssl                   1.1.1e                h0c8e037_0
pandas                    0.24.2             py27hc56fc5f_0
pip                       19.3.1                    py27_0
psycopg2                  2.8.4              py27hcfb25f9_0
python                    2.7.18                hfb89ab9_0
python-dateutil           2.8.2               pyhd3eb1b0_0
pytz                      2021.3              pyhd3eb1b0_0
scipy                     1.2.1              py27h4c3ab11_0
setuptools                44.0.0                    py27_0
six                       1.16.0              pyhd3eb1b0_1
sqlite                    3.30.1                h0c8e037_0
sqlparse                  0.4.1                       py_0
tk                        8.6.8                 h0c8e037_0
vc                        9                     h2eaa2aa_6
vs2008_runtime            9.00.30729.1          haa95532_6
wheel                     0.37.1              pyhd3eb1b0_0
wincertstore              0.2                py27hf04cefb_0
zlib                      1.2.11                h3cc03e0_3
```

Рисунок 2. Установленные пакеты

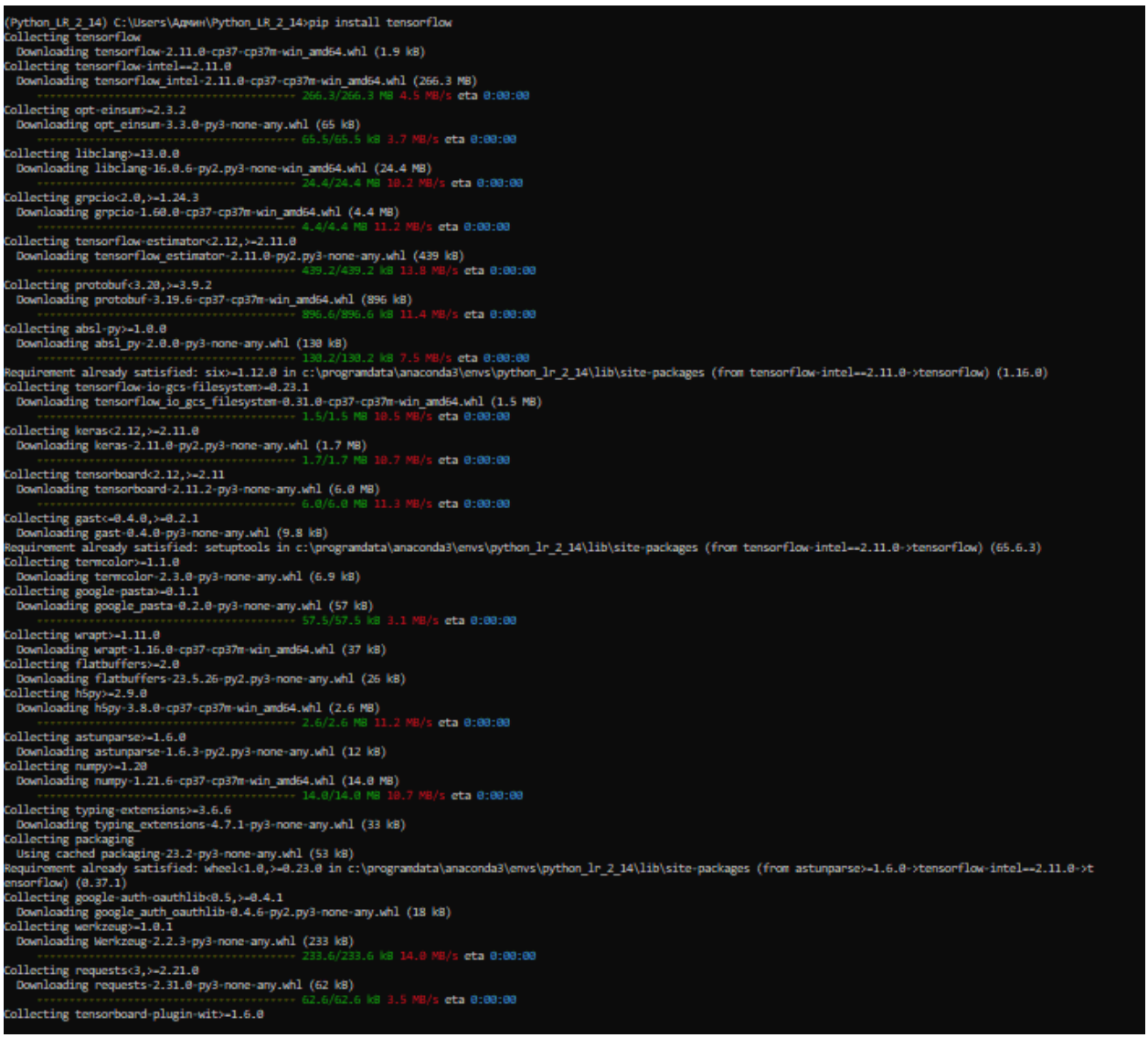

Рисунок 4. Процесс установки пакета TensorFlow с помощью менеджера пакетов pip

В файле requirements.txt обычно перечисляются все зависимости проекта, включая версии пакетов.

```
# This file may be used to create an environment using:
# $ conda create --name <env> --file <this file>
# platform: win-64
absl-py=2.0.0=pypi_0
asgiref=3.2.3=py_0
astunparse=1.6.3=pypi_0
blas=1.0=mkl
bottleneck=1.3.5=py37h080aedc_0
ca-certificates=2023.12.12=haa95532_0
cachetools=5.3.2=pypi_0
certifi=2022.12.7=py37haa95532_0
charset-normalizer=3.3.2=pypi_0
django=3.2=pyhd3eb1b0_0
flatbuffers=23.5.26=pypi_0
gast=0.4.0=pypi_0
google-auth=2.25.2=pypi_0
google-auth-oauthlib=0.4.6=pypi_0
```

```
google-pasta=0.2.0=pypi_0
grpcio=1.60.0=pypi_0
h5py=3.8.0=pypi_0
icc_rt=2022.1.0=h6049295_2
idna=3.6=pypi_0
importlib-metadata=6.7.0=pypi_0
intel-openmp=2022.0.0=haa95532_3663
keras=2.11.0=pypi_0
libclang=16.0.6=pypi_0
libpq=12.2=hb652d5d_1
markdown=3.4.4=pypi_0
markupsafe=2.1.3=pypi_0
mkl=2020.2=256
mkl-service=2.3.0=py37h196d8e1_0
mkl_fft=1.3.0=py37h46781fe_0
mkl_random=1.1.1=py37h47e9c7a_0
numexpr=2.7.3=py37hcbcaa1e_0
numpy=1.21.6=pypi_0
oauthlib=3.2.2=pypi_0
openssl=1.1.1w=h2bbff1b_0
opt-einsum=3.3.0=pypi_0
packaging=23.2=pypi_0
pandas=1.3.5=py37h6214cd6_0
pip=22.3.1=py37haa95532_0
protobuf=3.19.6=pypi_0
psycopg2=2.8.6=py37hcd4344a_1
pyasn1=0.5.1=pypi_0
pyasn1-modules=0.3.0=pypi_0
python=3.7.16=h6244533_0
python-dateutil=2.8.2=pyhd3eb1b0_0
pytz=2021.3=pyhd3eb1b0_0
requests=2.31.0=pypi_0
requests-oauthlib=1.3.1=pypi_0
rsa=4.9=pypi_0
scipy=1.6.2=py37h14eb087_0
setuptools=65.6.3=py37haa95532_0
six=1.16.0=pyhd3eb1b0_1
sqlite=3.41.2=h2bbff1b_0
sqlparse=0.4.1=py_0
tensorboard=2.11.2=pypi_0
tensorboard-data-server=0.6.1=pypi_0
tensorboard-plugin-wit=1.8.1=pypi_0
tensorflow=2.11.0=pypi_0
tensorflow-estimator=2.11.0=pypi_0
tensorflow-intel=2.11.0=pypi_0
tensorflow-io-gcs-filesystem=0.31.0=pypi_0
termcolor=2.3.0=pypi_0
typing-extensions=4.7.1=pypi_0
urllib3=2.0.7=pypi_0
vc=14.2=h21ff451_1
vs2008_runtime=9.00.30729.1=haa95532_6
vs2015_runtime=14.27.29016=h5e58377_2
werkzeug=2.2.3=pypi_0
wheel=0.37.1=pyhd3eb1b0_0
wincertstore=0.2=py37haa95532_2
wrapt=1.16.0=pypi_0
zipp=3.15.0=pypi_0
```

Файл environment.yml для использования с conda. Файл environment.yml обычно используется для создания виртуальных окружений с помощью conda.

```yaml
name: Python_LR_2_14
channels:
  - defaults
dependencies:
  - asgiref=3.2.3=py_0
  - blas=1.0=mkl
  - bottleneck=1.3.5=py37h080aedc_0
  - ca-certificates=2023.12.12=haa95532_0
  - certifi=2022.12.7=py37haa95532_0
  - django=3.2=pyhd3eb1b0_0
  - icc_rt=2022.1.0=h6049295_2
  - intel-openmp=2022.0.0=haa95532_3663
  - libpq=12.2=hb652d5d_1
  - mkl=2020.2=256
  - mkl-service=2.3.0=py37h196d8e1_0
  - mkl_fft=1.3.0=py37h46781fe_0
  - mkl_random=1.1.1=py37h47e9c7a_0
  - numexpr=2.7.3=py37hcbcaa1e_0
  - openssl=1.1.1w=h2bbff1b_0
  - pandas=1.3.5=py37h6214cd6_0
  - pip=22.3.1=py37haa95532_0
  - psycopg2=2.8.6=py37hcd4344a_1
  - python=3.7.16=h6244533_0
  - python-dateutil=2.8.2=pyhd3eb1b0_0
  - pytz=2021.3=pyhd3eb1b0_0
  - scipy=1.6.2=py37h14eb087_0
  - setuptools=65.6.3=py37haa95532_0
  - six=1.16.0=pyhd3eb1b0_1
  - sqlite=3.41.2=h2bbff1b_0
  - sqlparse=0.4.1=py_0
  - vc=14.2=h21ff451_1
  - vs2008_runtime=9.00.30729.1=haa95532_6
  - vs2015_runtime=14.27.29016=h5e58377_2
  - wheel=0.37.1=pyhd3eb1b0_0
  - wincertstore=0.2=py37haa95532_2
  - pip:
      - absl-py==2.0.0
      - astunparse==1.6.3
      - cachetools==5.3.2
      - charset-normalizer==3.3.2
      - flatbuffers==23.5.26
      - gast==0.4.0
      - google-auth==2.25.2
      - google-auth-oauthlib==0.4.6
      - google-pasta==0.2.0
      - grpcio==1.60.0
      - h5py==3.8.0
      - idna==3.6
      - importlib-metadata==6.7.0
      - keras==2.11.0
      - libclang==16.0.6
      - markdown==3.4.4
      - markupsafe==2.1.3
      - numpy==1.21.6
      - oauthlib==3.2.2
      - opt-einsum==3.3.0
      - packaging==23.2
      - protobuf==3.19.6
      - pyasn1==0.5.1
      - pyasn1-modules==0.3.0
      - requests==2.31.0
      - requests-oauthlib==1.3.1
      - rsa==4.9
      - tensorboard==2.11.2
```

```
      - tensorboard-data-server==0.6.1
      - tensorboard-plugin-wit==1.8.1
      - tensorflow==2.11.0
      - tensorflow-estimator==2.11.0
      - tensorflow-intel==2.11.0
      - tensorflow-io-gcs-filesystem==0.31.0
      - termcolor==2.3.0
      - typing-extensions==4.7.1
      - urllib3==2.0.7
      - werkzeug==2.2.3
      - wrapt==1.16.0
      - zipp==3.15.0
prefix: C:\ProgramData\anaconda3\envs\Python_LR_2_14
```

Контрольные вопросы:

1. Существует так называемый Python Package Index (PyPI) – это репозиторий, открытый для всех Python разработчиков, в нем вы можете найти пакеты для решения практически любых задач. Там также есть возможность выкладывать свои пакеты. Для скачивания и установки используется специальная утилита, которая называется pip.

2. При развертывании современной версии Python (начиная с Python 2.7.9 и Python 3.4), pip устанавливается автоматически. Но если, по какой-то причине, pip не установлен на вашем ПК, то сделать это можно вручную. Будем считать, что Python у вас уже установлен, теперь необходимо установить pip. Для того, чтобы это сделать, скачайте скрипт get-pip.py $ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py и выполните его. $ python get-pip.py При этом, вместе с pip будут установлены setuptools и wheels. Setuptools – это набор инструментов для построения пакетов Python. Wheels – это формат дистрибутива для пакета Python.

3. По умолчанию менеджер пакетов pip скачивает пакеты из Python Package Index (PyPI).

4. pip install ProjectName

5. pip install ProjectName==*

6. pip install -e git+https://gitrepo.com/ProjectName.git

7. pip install ./dist/ProjectName.tar.gz

8. pip uninstall ProjectName

9. pip install --upgrade ProjectName

10. pip list

11. В системе для интерпретатора Python может быть установлена глобально только одна версия пакета. Это порождает ряд проблем: проблема обратной совместимости и проблема коллективной разработки. Получается, что для каждого проекта нужна своя "песочница", которая изолирует зависимости. Такая "песочница" придумана и называется "виртуальным окружением" или "виртуальной средой".

12. Создаём через утилиту новое виртуальное окружение в отдельной папке для выбранной версии интерпретатора Python. − Активируем ранее созданное виртуального окружения для работы. − Работаем в виртуальном окружении, а именно управляем пакетами используя pip и запускаем выполнение кода. − Деактивируем после окончания работы виртуальное окружение. − Удаляем папку с виртуальным окружением, если оно нам больше не нужно.

13. Для создания виртуального окружения достаточно дать команду в формате: python3 -m venv Обычно папку для виртуального окружения называют env или venv. В описании команды выше явно указан интерпретатор версии 3.x. Под Windows и некоторыми другими операционными системами это будет просто python. Чтобы активировать виртуальное окружение нужно: $ source env/bin/activate В Windows мы вызываем скрипт активации напрямую. > env\\Scripts\\activate Чтобы переключиться с одного окружения на другое нам нужно выполнить команду деактивации и команду активации другого виртуального окружения, например, так: $ deactivate

14. Для начала пакет нужно установить. Установку можно выполнить командой: # Для python 3 python3 -m pip install virtualenv # Для единственного python python -m pip install virtualenv Создание виртуального окружения с утилитой virtualenv отличается от стандартного. Например, создание в текущей папке виртуального окружения для интерпретатора доступного через команду python3 с названием папки окружения env: virtualenv -p python3 env Активация и деактивация такая же, как у стандартной утилиты Python.

15. pipenv install — Создание виртуального окружения pipenv install — Установка определённого пакета и добавление его в Pipfile. pipenv uninstall — Удаление установленного пакета и его исключение из Pipfile. зависимостей. pipenv shell — Активация виртуального окружения.

16. Просмотреть список зависимостей мы можем командой: pip freeze Что бы его сохранить, нужно перенаправить вывод команды в файл: pip freeze > requirements.txt Имя файла хранения зависимостей requirements.txt выбрано не зря. Оно является стандартной договоренностью и используется некоторыми утилитами автоматически. Установка пакетов из файла зависимостей в новом виртуальном окружении так же выполняется одной командой: pip install -r requirements.txt

17. Основная проблема заключается в том, что pip , easy_install и virtualenv ориентированы на Python. Эти инструменты игнорируют библиотеки зависимостей, реализованные с использованием других языков. Например, XSLT , HDF5 , MKL и другие, которые не имеют setup.py в исходном коде и не устанавливают файлы в директорию site-packages. Conda же способна управлять пакетами как для Python, так и для C/ C++, R, Ruby, Lua, Scala и других. Conda устанавливает двоичные файлы, поэтому работу по компиляции пакета самостоятельно выполнять не требуется (по сравнению с pip). Существуют также некоторые различия, если вы заинтересованы в создании собственных пакетов. Например, pip создан на основе setuptools, тогда как conda использует свой собственный формат, который имеет некоторые преимущества (например, статическая компиляция пакета).

18. Anaconda и Miniconda.

19. Начиная проект, создайте чистую директорию и дайте ей понятное короткое имя. Для Linux это будет соответствовать набору команд: mkdir $PROJ_NAME cd $PROJ_NAME touch README.md main.py Создайте чистое conda-окружение с таким же именем: conda create -n $PROJ_NAME python=3.7

20. conda activate $PROJ_NAME conda install $PACKAGE_NAME

21. conda deactivate conda remove -n $PACKAGE_NAME

22. Файл environment.yml позволит воссоздать окружение в любой нужный момент. conda env export > environment.yml

23. Как создать виртуальное окружение conda с помощью файла environment.yml

24. Необходимо установить Anaconda или Miniconda. В Pycharm необходимо настроить интерпретатор Python: Нужно перейти в File > Settings (для Windows/Linux) или PyCharm > Preferences (для macOS). В левой части окна настроек выбрать Project: ваш_проект > Python Interpreter. Нажать на шестерёнку справа от списка интерпретаторов и выбрать Add. В открывшемся окне добавления интерпретатора выбрать Conda Environment. Можно либо создать новое окружение, выбрав New environment, либо использовать существующее, выбрав Existing environment. Создание нового окружения Conda: Необходимо указать имя окружения, версию Python и нажать кнопку OK. PyCharm автоматически создаст новое окружение Conda и установит в него выбранную версию Python. Использование существующего окружения Conda: Нужно нажать на кнопку с тремя точками и найти путь к существующему окружению Conda. Активация окружения Conda: При использовании терминала в PyCharm окружение Conda должно активироваться автоматически. Если этого не произошло, его можно активировать вручную, введя команду conda activate имя_окружения в терминале. Работа с проектом: После настройки окружения Conda можно работать с проектом в PyCharm, как обычно.

25. Чтобы пользователи, которые скачивают какие-либо программы, скрипты, модули могли без проблем посмотреть, какие пакеты им нужно установить дополнительно для корректной работы. За описание о наличии каких-либо пакетов в среде как раз и отвечают файлы requirements.txt и environment.yml.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе с менеджером пакетов pip и виртуальными окружениями с помощью языка программирования Python версии 3.x