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 ЛАБОРАТОРНОЙ РАБОТЕ №17
дисциплины «Программирование на Python»

Выполнил:
Данилецкий Дмитрий Витальевич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и вычислительная техника»,
направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем», очная форма обучения

_____
(подпись)

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

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: Установка пакетов в Python. Виртуальные окружения

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

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.

2. Дополнил файл .gitignore необходимыми правилами.

3. Организовал созданный репозиторий в соответствие с моделью ветвления git-flow.

4. Создал виртуальное окружение Anaconda с именем репозитория.

```
(base) C:\Users\slime>cd LAB17

(base) C:\Users\slime\LAB17>mkdir ocrconda

(base) C:\Users\slime\LAB17>cd ocrconda

(base) C:\Users\slime\LAB17\ocrconda>touch README.md main.py
"touch" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

(base) C:\Users\slime\LAB17\ocrconda>conda create -n LAB17 python=3.11
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 23.11.0

Please update conda by running

    $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

     conda install conda=23.11.0
```

Рисунок 1. Создание виртуального окружения Anaconda

```
Proceed ([y]/n)? y


Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate LAB17
#
# To deactivate an active environment, use
#
#     $ conda deactivate


(base) C:\Users\slime\LAB17\ocrconda>conda activate LAB17
"conda" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

(base) C:\Users\slime\LAB17\ocrconda>conda activate LAB17

(LAB17) C:\Users\slime\LAB17\ocrconda>
```

Рисунок 2. Завершение создания и активация виртуального окружения Anaconda

5. Установил в виртуальное окружение следующие пакеты: pip, NumPy, Pandas, SciPy.

```
The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    intel-openmp-2023.1.0      |   h59b6b97_46320         2.7 MB
    mkl-2023.1.0               |   h6b88ed4_46358       155.9 MB
    numexpr-2.8.7              |  py311h1fcbade_0         160 KB
    numpy-1.26.2               |  py311hdab7c0b_0          11 KB
    numpy-base-1.26.2          |  py311hd01c5d8_0         7.2 MB
    pandas-2.1.4               |  py311hf62ec03_0        13.6 MB
    scipy-1.11.4               |  py311hc1ccb85_0        20.9 MB
    ------------------------------------------------------------
                                           Total:       200.4 MB

The following NEW packages will be INSTALLED:

  blas               pkgs/main/win-64::blas-1.0-mkl
  bottleneck         pkgs/main/win-64::bottleneck-1.3.5-py311h5bb9823_0
  icc_rt             pkgs/main/win-64::icc_rt-2022.1.0-h6049295_2
  intel-openmp       pkgs/main/win-64::intel-openmp-2023.1.0-h59b6b97_46320
  mkl                pkgs/main/win-64::mkl-2023.1.0-h6b88ed4_46358
  mkl-service        pkgs/main/win-64::mkl-service-2.4.0-py311h2bbff1b_1
  mkl_fft            pkgs/main/win-64::mkl_fft-1.3.8-py311h2bbff1b_0
  mkl_random         pkgs/main/win-64::mkl_random-1.2.4-py311h59b6b97_0
  numexpr            pkgs/main/win-64::numexpr-2.8.7-py311h1fcbade_0
  numpy              pkgs/main/win-64::numpy-1.26.2-py311hdab7c0b_0
  numpy-base         pkgs/main/win-64::numpy-base-1.26.2-py311hd01c5d8_0
  pandas             pkgs/main/win-64::pandas-2.1.4-py311hf62ec03_0
  python-dateutil    pkgs/main/noarch::python-dateutil-2.8.2-pyhd3eb1b0_0
  python-tzdata      pkgs/main/noarch::python-tzdata-2023.3-pyhd3eb1b0_0
  pytz               pkgs/main/win-64::pytz-2023.3.post1-py311haa95532_0
  scipy              pkgs/main/win-64::scipy-1.11.4-py311hc1ccb85_0
  six                pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1
  tbb                pkgs/main/win-64::tbb-2021.8.0-h59b6b97_0


Proceed ([y]/n)? y


Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(LAB17) C:\Users\slime\LAB17\ocrconda>
```

Рисунок 3. Установка пакетов в виртуальное окружение

6. Попробовал установить менеджером пакетов conda пакет TensorFlow. При этом возникла ошибка. Проблема заключается в конфликте версий Python при установке TensorFlow. Ваша версия Python в среде Conda - 3.11, и TensorFlow ищет совместимую версию в диапазоне от 3.5 до 3.10. Поскольку TensorFlow не поддерживает Python 3.11, возникает ошибка.

```
(LAB17) C:\Users\slime\LAB17\ocrconda>conda install TensorFlow
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Solving environment: unsuccessful attempt using repodata from current_repodata.json, retrying with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Solving environment: |
Found conflicts! Looking for incompatible packages.
This can take several minutes.  Press CTRL-C to abort.
failed

UnsatisfiableError: The following specifications were found
to be incompatible with the existing python installation in your environment:

Specifications:

  - tensorflow -> python[version='3.10.*|3.9.*|3.8.*|3.7.*|3.6.*|3.5.*']

Your python: python=3.11

If python is on the left-most side of the chain, that's the version you've asked for.
When python appears to the right, that indicates that the thing on the left is somehow
not available for the python version you are constrained to. Note that conda will not
change your python version to a different minor version unless you explicitly specify
that.
```

Рисунок 4. Ошибка установки пакета TensorFlow

7. Установил пакет TensorFlow с помощью менеджера пакетов pip.

```
     ---------------------------------------- 24.4/24.4 MB 6.7 MB/s eta 0:00:00
Downloading ml_dtypes-0.2.0-cp311-cp311-win_amd64.whl (938 kB)
     ---------------------------------------- 938.7/938.7 kB 11.8 MB/s eta 0:00:00
Downloading tensorboard-2.15.1-py3-none-any.whl (5.5 MB)
     ---------------------------------------- 5.5/5.5 MB 10.7 MB/s eta 0:00:00
Downloading protobuf-4.23.4-cp310-abi3-win_amd64.whl (422 kB)
     ---------------------------------------- 422.5/422.5 kB 13.3 MB/s eta 0:00:00
Downloading tensorflow_estimator-2.15.0-py2.py3-none-any.whl (441 kB)
     ---------------------------------------- 442.0/442.0 kB 9.2 MB/s eta 0:00:00
Downloading termcolor-2.4.0-py3-none-any.whl (7.7 kB)
Downloading typing_extensions-4.9.0-py3-none-any.whl (32 kB)
Downloading wrapt-1.14.1-cp311-cp311-win_amd64.whl (35 kB)
Downloading packaging-23.2-py3-none-any.whl (53 kB)
     ---------------------------------------- 53.0/53.0 kB ? eta 0:00:00
Downloading google_auth-2.25.2-py2.py3-none-any.whl (184 kB)
     ---------------------------------------- 184.2/184.2 kB 11.6 MB/s eta 0:00:00
Downloading google_auth_oauthlib-1.2.0-py2.py3-none-any.whl (24 kB)
Downloading Markdown-3.5.1-py3-none-any.whl (102 kB)
     ---------------------------------------- 102.2/102.2 kB 6.1 MB/s eta 0:00:00
Downloading requests-2.31.0-py3-none-any.whl (62 kB)
     ---------------------------------------- 62.6/62.6 kB 3.3 MB/s eta 0:00:00
Downloading tensorboard_data_server-0.7.2-py3-none-any.whl (2.4 kB)
Downloading werkzeug-3.0.1-py3-none-any.whl (226 kB)
     ---------------------------------------- 226.7/226.7 kB 7.0 MB/s eta 0:00:00
Downloading cachetools-5.3.2-py3-none-any.whl (9.3 kB)
Downloading certifi-2023.11.17-py3-none-any.whl (162 kB)
     ---------------------------------------- 162.5/162.5 kB 10.2 MB/s eta 0:00:00
Downloading charset_normalizer-3.3.2-cp311-cp311-win_amd64.whl (99 kB)
     ---------------------------------------- 99.9/99.9 kB 5.6 MB/s eta 0:00:00
Downloading idna-3.6-py3-none-any.whl (61 kB)
     ---------------------------------------- 61.6/61.6 kB 3.2 MB/s eta 0:00:00
Downloading MarkupSafe-2.1.3-cp311-cp311-win_amd64.whl (17 kB)
Downloading urllib3-2.1.0-py3-none-any.whl (104 kB)
     ---------------------------------------- 104.6/104.6 kB 5.9 MB/s eta 0:00:00
Downloading pyasn1-0.5.1-py2.py3-none-any.whl (84 kB)
     ---------------------------------------- 84.9/84.9 kB 4.7 MB/s eta 0:00:00
Installing collected packages: libclang, flatbuffers, wrapt, urllib3, typing-extensions, termcolor, tensorflow-io-gcs-filesystem, tensorflow-estima
tor, tensorboard-data-server, pyasn1, protobuf, packaging, opt-einsum, oauthlib, ml-dtypes, MarkupSafe, markdown, keras, idna, h5py, grpcio, google
-pasta, gast, charset-normalizer, certifi, cachetools, astunparse, absl-py, werkzeug, rsa, requests, pyasn1-modules, requests-oauthlib, google-auth
, google-auth-oauthlib, tensorboard, tensorflow-intel, tensorflow
Successfully installed MarkupSafe-2.1.3 absl-py-2.0.0 astunparse-1.6.3 cachetools-5.3.2 certifi-2023.11.17 charset-normalizer-3.3.2 flatbuffers-23.
5.26 gast-0.5.4 google-auth-2.25.2 google-auth-oauthlib-1.2.0 google-pasta-0.2.0 grpcio-1.60.0 h5py-3.10.0 idna-3.6 keras-2.15.0 libclang-16.0.6 ma
rkdown-3.5.1 ml-dtypes-0.2.0 oauthlib-3.2.2 opt-einsum-3.3.0 packaging-23.2 protobuf-4.23.4 pyasn1-0.5.1 pyasn1-modules-0.3.0 requests-2.31.0 reque
sts-oauthlib-1.3.1 rsa-4.9 tensorboard-2.15.1 tensorboard-data-server-0.7.2 tensorflow-2.15.0 tensorflow-estimator-2.15.0 tensorflow-intel-2.15.0 t
ensorflow-io-gcs-filesystem-0.31.0 termcolor-2.4.0 typing-extensions-4.9.0 urllib3-2.1.0 werkzeug-3.0.1 wrapt-1.14.1

(LAB17) C:\Users\slime\LAB17\ocrconda>
```

Рисунок 5. Успешная установка пакета TensorFlow

8. Сформировал файлы requirements.txt и environment.yml. В requirements.txt хранится список всех зависимостей проекта и их версий (установленных пакетов до выполнения команды). Environment.yml предназначен для определения и создания виртуальной среды, которая включает в себя не только зависимости Python (библиотеки), но и другие параметры окружения, такие как версия Python, пакеты системного уровня и т. д.

```
(LAB17) C:\Users\slime\LAB17\ocrconda>conda env export > environment.yml

(LAB17) C:\Users\slime\LAB17\ocrconda>pip freeze > requirements.txt

(LAB17) C:\Users\slime\LAB17\ocrconda>
```

Рисунок 6. Создание файлов requirements.txt и environment.yml

```
Файл  Правка  Формат  Вид  Справка
absl-py==2.0.0
astunparse==1.6.3
Bottleneck @ file:///C:/ci_311/bottleneck_1676500016583/work
cachetools==5.3.2
certifi==2023.11.17
charset-normalizer==3.3.2
flatbuffers==23.5.26
gast==0.5.4
google-auth==2.25.2
google-auth-oauthlib==1.2.0
google-pasta==0.2.0
grpcio==1.60.0
h5py==3.10.0
idna==3.6
keras==2.15.0
libclang==16.0.6
Markdown==3.5.1
MarkupSafe==2.1.3
mkl-fft @ file:///C:/b/abs_19i1y8ykas/croot/mkl_fft_1695058226480/work
mkl-random @ file:///C:/b/abs_edwkj1_o69/croot/mkl_random_1695059866750/work
mkl-service==2.4.0
ml-dtypes==0.2.0
numexpr @ file:///C:/b/abs_5fucrty5dc/croot/numexpr_1696515448831/work
numpy @ file:///C:/b/abs_7267ja_mqz/croot/numpy_and_numpy_base_1701295083047/work/dist/numpy-1.26.2-cp311-cp311-win_amd64.whl#sha256=f4e43aff0bd2ec1e69abffbe07459b4dcf6ae207942d293dd3c14112aa00d3
oauthlib==3.2.2
opt-einsum==3.3.0
packaging==23.2
pandas @ file:///C:/b/abs_fej9bi0gew/croot/pandas_1702318041921/work/dist/pandas-2.1.4-cp311-cp311-win_amd64.whl#sha256=d3609b7cc3e3c4d99ad640a4b8e710ba93ccf967ab8e5245b91033e0200f9286
protobuf==4.23.4
pyasn1==0.5.1
pyasn1-modules==0.3.0
python-dateutil @ file:///tmp/build/80754af9/python-dateutil_1626374649649/work
pytz @ file:///C:/b/abs_19q3ljkez4/croot/pytz_1695131651401/work
requests==2.31.0
requests-oauthlib==1.3.1
rsa==4.9
scipy==1.11.4
six @ file:///tmp/build/80754af9/six_1644875935023/work
tensorboard==2.15.1
tensorboard-data-server==0.7.2
tensorflow==2.15.0
tensorflow-estimator==2.15.0
tensorflow-intel==2.15.0
tensorflow-io-gcs-filesystem==0.31.0
termcolor==2.4.0
typing_extensions==4.9.0
tzdata @ file:///croot/python-tzdata_1690578112552/work
urllib3==2.1.0
Werkzeug==3.0.1
wrapt==1.14.1
```

Рисунок 7. Содержимое файла requirements.txt

environment.yml | form.html | index.html

: > Users > slime > LAB17 > ocrconda > ! environment.yml

```
name: LAB17
channels:
  - defaults
dependencies:
  - blas=1.0=mkl
  - bottleneck=1.3.5=py311h5bb9823_0
  - bzip2=1.0.8=he774522_0
  - ca-certificates=2023.12.12=haa95532_0
  - icc_rt=2022.1.0=h6049295_2
  - intel-openmp=2023.1.0=h59b6b97_46320
  - libffi=3.4.4=hd77b12b_0
  - mkl=2023.1.0=h6b88ed4_46358
  - mkl-service=2.4.0=py311h2bbff1b_1
  - mkl_fft=1.3.8=py311h2bbff1b_0
  - mkl_random=1.2.4=py311h59b6b97_0
  - numexpr=2.8.7=py311h1fcbade_0
  - numpy=1.26.2=py311hdab7c0b_0
  - numpy-base=1.26.2=py311hd01c5d8_0
  - openssl=3.0.12=h2bbff1b_0
  - pandas=2.1.4=py311hf62ec03_0
  - pip=23.3.1=py311haa95532_0
  - python=3.11.5=he1021f5_0
  - python-dateutil=2.8.2=pyhd3eb1b0_0
  - python-tzdata=2023.3=pyhd3eb1b0_0
  - pytz=2023.3.post1=py311haa95532_0
  - scipy=1.11.4=py311hc1ccb85_0
  - setuptools=68.2.2=py311haa95532_0
  - six=1.16.0=pyhd3eb1b0_1
  - sqlite=3.41.2=h2bbff1b_0
  - tbb=2021.8.0=h59b6b97_0
  - tk=8.6.12=h2bbff1b_0
  - tzdata=2023c=h04d1e81_0
  - vc=14.2=h21ff451_1
  - vs2015_runtime=14.27.29016=h5e58377_2
  - wheel=0.41.2=py311haa95532_0
  - xz=5.4.5=h8cc25b3_0
  - zlib=1.2.13=h8cc25b3_0
  - pip:
      - absl-py==2.0.0
      - astunparse==1.6.3
      - cachetools==5.3.2
      - certifi==2023.11.17
      - charset-normalizer==3.3.2
      - flatbuffers==23.5.26
      - gast==0.5.4
      - google-auth==2.25.2
      - google-auth-oauthlib==1.2.0
      - google-pasta==0.2.0
      - grpcio==1.60.0
      - h5py==3.10.0
      - idna==3.6
      - keras==2.15.0
      - libclang==16.0.6
      - markdown==3.5.1
      - markupsafe==2.1.3
      - ml-dtypes==0.2.0
      - oauthlib==3.2.2
      - opt-einsum==3.3.0
      - packaging==23.2
      - protobuf==4.23.4
      - pyasn1==0.5.1
      - pyasn1-modules==0.3.0
      - requests==2.31.0
      - requests-oauthlib==1.3.1
      - rsa==4.9
      - tensorboard==2.15.1
      - tensorboard-data-server==0.7.2
      - tensorflow==2.15.0
      - tensorflow-estimator==2.15.0
      - tensorflow-intel==2.15.0
      - tensorflow-io-gcs-filesystem==0.31.0
      - termcolor==2.4.0
      - typing-extensions==4.9.0
      - urllib3==2.1.0
      - werkzeug==3.0.1
      - wrapt==1.14.1
prefix: C:\Users\slime\anaconda3\envs\LAB17

```

Рисунок 8. Содержимое файла environment.yml

Контрольные вопросы

1. Каким способом можно установить пакет Python, не входящий в стандартную библиотеку?

Для установки пакета Python, не входящего в стандартную библиотеку, можно воспользоваться менеджером пакетов pip. Например, для установки пакета "requests" выполните команду: pip install requests

2. Как осуществить установку менеджера пакетов pip?

Для установки менеджера пакетов pip, обычно он устанавливается вместе с Python. Если он не установлен, можно воспользоваться инструкцией по установке pip для вашей операционной системы.

3. Откуда менеджер пакетов pip по умолчанию устанавливает пакеты?

По умолчанию pip устанавливает пакеты из Python Package Index (PyPI), но также может устанавливать их из других источников, таких как Git репозитории.

4. Как установить последнюю версию пакета с помощью pip?

Для установки последней версии пакета с помощью pip, используйте команду: pip install --upgrade package_name

5. Как установить заданную версию пакета с помощью pip?

Для установки заданной версии пакета с помощью pip, используйте команду: pip install package_name==version_number

6. Как установить пакет из git репозитория (в том числе GitHub) с помощью pip?

Для установки пакета из git репозитория с помощью pip, используйте команду: pip install git+https://github.com/username/repository.git

7. Как установить пакет из локальной директории с помощью pip?

Для установки пакета из локальной директории с помощью pip, используйте команду: pip install /path/to/local/directory

8. Как удалить установленный пакет с помощью pip?

Для удаления установленного пакета с помощью pip, используйте команду: pip uninstall package_name

9. Как обновить установленный пакет с помощью pip?

Для обновления установленного пакета с помощью pip, используйте команду: pip install --upgrade package_name

10. Как отобразить список установленных пакетов с помощью pip?

Для отображения списка установленных пакетов с помощью pip, используйте команду: pip list

11. Каковы причины появления виртуальных окружений в языке Python?

Виртуальные окружения в Python используются для изоляции проектов и их зависимостей, чтобы избежать конфликтов между различными версиями пакетов.

12. Каковы основные этапы работы с виртуальными окружениями?

Основные этапы работы с виртуальными окружениями включают создание, активацию, деактивацию и удаление виртуальных окружений.

13. Как осуществляется работа с виртуальными окружениями с помощью venv?

Для работы с виртуальными окружениями с помощью venv, используйте стандартную библиотеку Python для создания и управления виртуальными окружениями.

14. Как осуществляется работа с виртуальными окружениями с помощью virtualenv?

Virtualenv предоставляет инструменты для создания изолированных виртуальных окружений Python.

15. Изучите работу с виртуальными окружениями pipenv. Как осущестляется работа с виртуальными окружениями pipenv?

Pipenv предоставляет удобный способ управления зависимостями и виртуальными окружениями для проектов Python.

16. Каково назначение файла requirements.txt? Как создать этот файл? Какой он имеет формат?

Файл requirements.txt используется для хранения списка зависимостей проекта, что позволяет легко установить их на другой системе. Файл создается вручную и содержит список пакетов и их версий.

17. В чем преимущества пакетного менеджера conda по сравнению с пакетным менеджером pip?

Conda позволяет управлять не только Python-пакетами, но и библиотеками, написанными на других языках. Он также умеет

устанавливать библиотеки, которые содержат бинарные зависимости, что делает его более гибким по сравнению с pip.

18. В какие дистрибутивы Python входит пакетный менеджер conda?

Conda входит в дистрибутив Anaconda и Miniconda, которые предоставляют широкий выбор пакетов для научных вычислений и анализа данных.

19. Как создать виртуальное окружение conda?

Для создания виртуального окружения с помощью conda используется команда conda create --name myenv для создания нового окружения с именем "myenv".

20. Как активировать и установить пакеты в виртуальное окружение conda?

Для активации виртуального окружения conda используйте команду conda activate myenv, а для установки пакетов в это окружение используйте conda install package_name.

21. Как деактивировать и удалить виртуальное окружение conda?

Для деактивации виртуального окружения conda используйте команду conda deactivate, а для удаления окружения используйте conda remove --name myenv --all.

22. Каково назначение файла environment.yml ? Как создать этот файл?

Файл environment.yml используется для описания окружения conda, включая список пакетов и их версий. Этот файл можно создать вручную, указав необходимые пакеты и их версии, или сгенерировать автоматически с помощью команды conda env export > environment.yml.

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

Для создания виртуального окружения conda с использованием файла environment.yml, выполните команду conda env create -f environment.yml.

24. Самостоятельно изучите средства IDE PyCharm для работы с виртуальными окружениями conda. Опишите порядок работы с виртуальными окружениями conda в IDE PyCharm.

В PyCharm можно работать с виртуальными окружениями conda, создавая и активируя их через интерфейс пользователя. Для этого необходимо установить плагин Conda, после чего можно создавать, активировать и управлять виртуальными окружениями через интерфейс PyCharm.

25. Почему файлы requirements.txt и environment.yml должны храниться в репозитории git?

Файлы requirements.txt и environment.yml должны храниться в репозитории git, чтобы другие разработчики могли легко воссоздать окружение проекта на своих системах. Это позволяет обеспечить консистентность окружения и упростить процесс развертывания проекта

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