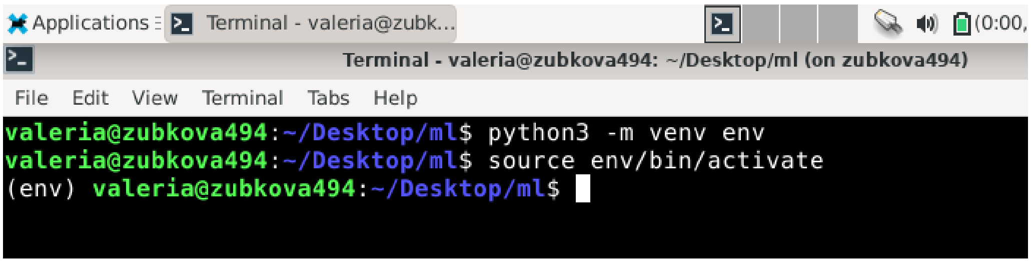
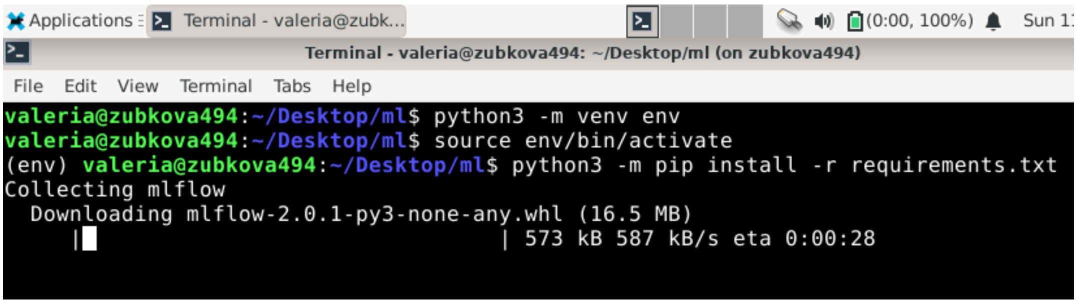
Предмет	Управление и автоматизация БД
Курс	4
Семестр	1
Работа	Практическое занятие № 37
Группа	494
Фамилия	Зубкова
Имя	Валерия
Отчество	Геннадьевна

Отчёт


Applications
Terminal - valeria@zubk...
(0:00,
Terminal - valeria@zubkova494: ~/Desktop/ml (on zubkova494)
File Edit View Terminal Tabs Help

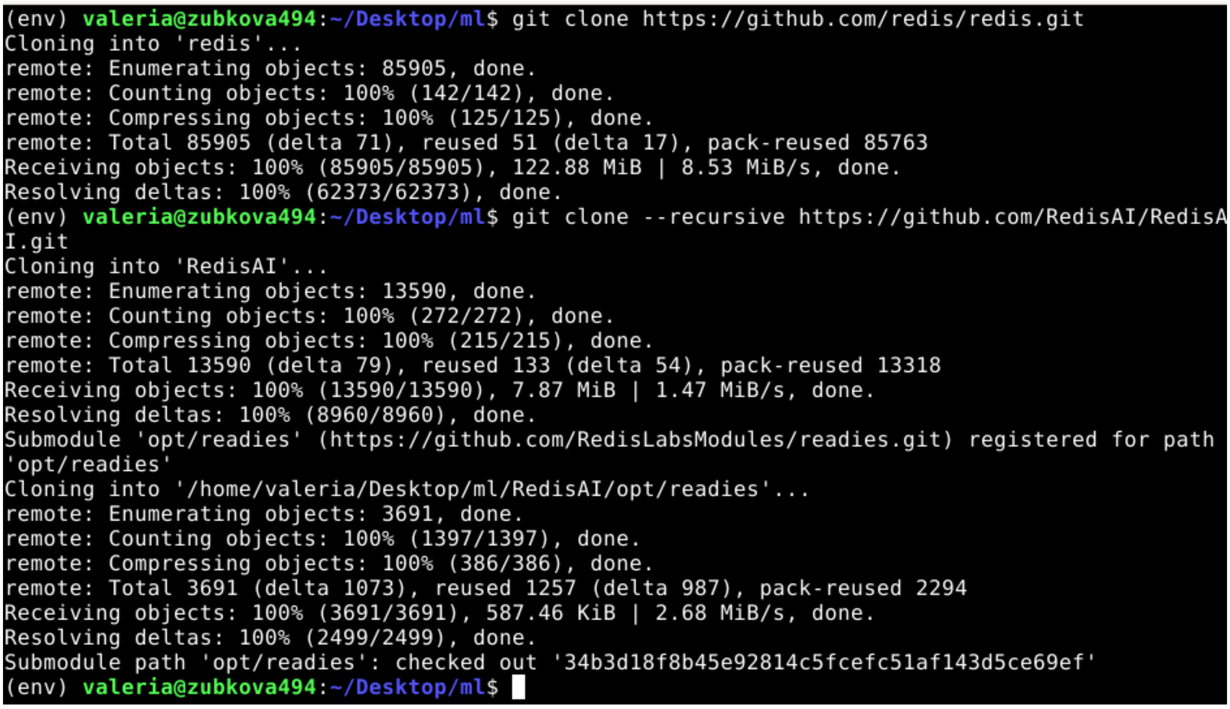
```
valeria@zubkova494:~/Desktop/ml$ python3 -m venv env
valeria@zubkova494:~/Desktop/ml$ source env/bin/activate
(env) valeria@zubkova494:~/Desktop/ml$ 
```

Рисунок 1 - Создание папки ml и виртуальной среды env, а ее активация


Applications
Terminal - valeria@zubk...
(0:00, 100%)
Sun 1
Terminal - valeria@zubkova494: ~/Desktop/ml (on zubkova494)
File Edit View Terminal Tabs Help

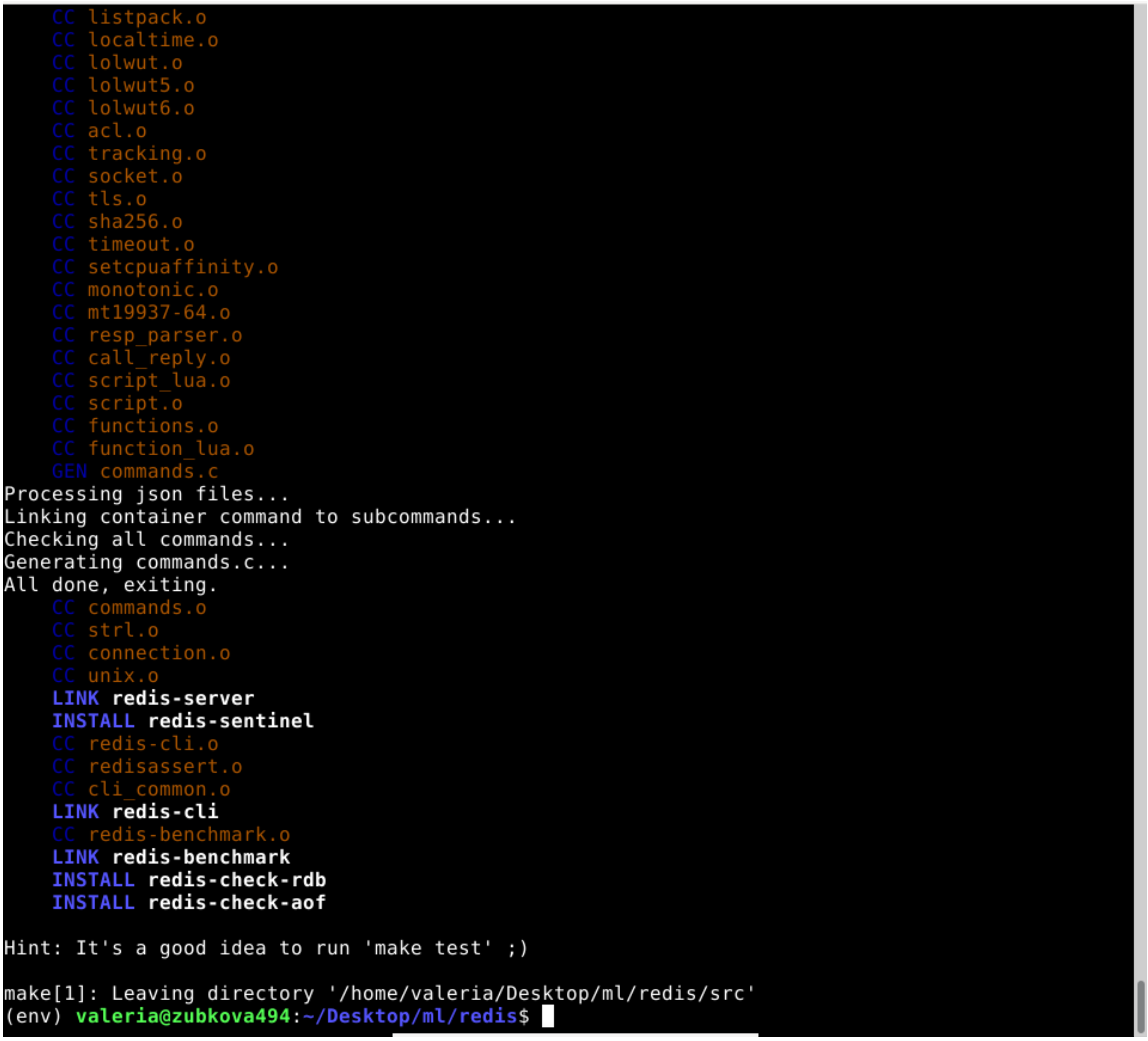
```
valeria@zubkova494:~/Desktop/ml$ python3 -m venv env
valeria@zubkova494:~/Desktop/ml$ source env/bin/activate
(env) valeria@zubkova494:~/Desktop/ml$ python3 -m pip install -r requirements.txt
Collecting mlflow
  Downloading mlflow-2.0.1-py3-none-any.whl (16.5 MB)
     |                                | 573 kB 587 kB/s eta 0:00:28
```

Рисунок 2 - Установка пакетов из файла requirements.txt



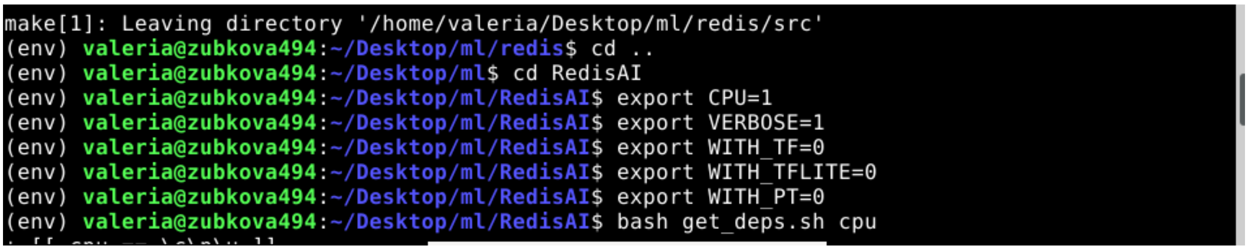
```
(env) valeria@zubkova494:~/Desktop/ml$ git clone https://github.com/redis/redis.git
Cloning into 'redis'...
remote: Enumerating objects: 85905, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (125/125), done.
remote: Total 85905 (delta 71), reused 51 (delta 17), pack-reused 85763
Receiving objects: 100% (85905/85905), 122.88 MiB | 8.53 MiB/s, done.
Resolving deltas: 100% (62373/62373), done.
(env) valeria@zubkova494:~/Desktop/ml$ git clone --recursive https://github.com/RedisAI/RedisA
I.git
Cloning into 'RedisAI'...
remote: Enumerating objects: 13590, done.
remote: Counting objects: 100% (272/272), done.
remote: Compressing objects: 100% (215/215), done.
remote: Total 13590 (delta 79), reused 133 (delta 54), pack-reused 13318
Receiving objects: 100% (13590/13590), 7.87 MiB | 1.47 MiB/s, done.
Resolving deltas: 100% (8960/8960), done.
Submodule 'opt/readies' (https://github.com/RedisLabsModules/readies.git) registered for path
'opt/readies'
Cloning into '/home/valeria/Desktop/ml/RedisAI/opt/readies'...
remote: Enumerating objects: 3691, done.
remote: Counting objects: 100% (1397/1397), done.
remote: Compressing objects: 100% (386/386), done.
remote: Total 3691 (delta 1073), reused 1257 (delta 987), pack-reused 2294
Receiving objects: 100% (3691/3691), 587.46 KiB | 2.68 MiB/s, done.
Resolving deltas: 100% (2499/2499), done.
Submodule path 'opt/readies': checked out '34b3d18f8b45e92814c5fcefc51af143d5ce69ef'
(env) valeria@zubkova494:~/Desktop/ml$ 
```

Рисунок 3 – скачиваем redis и redisai из репозиториев



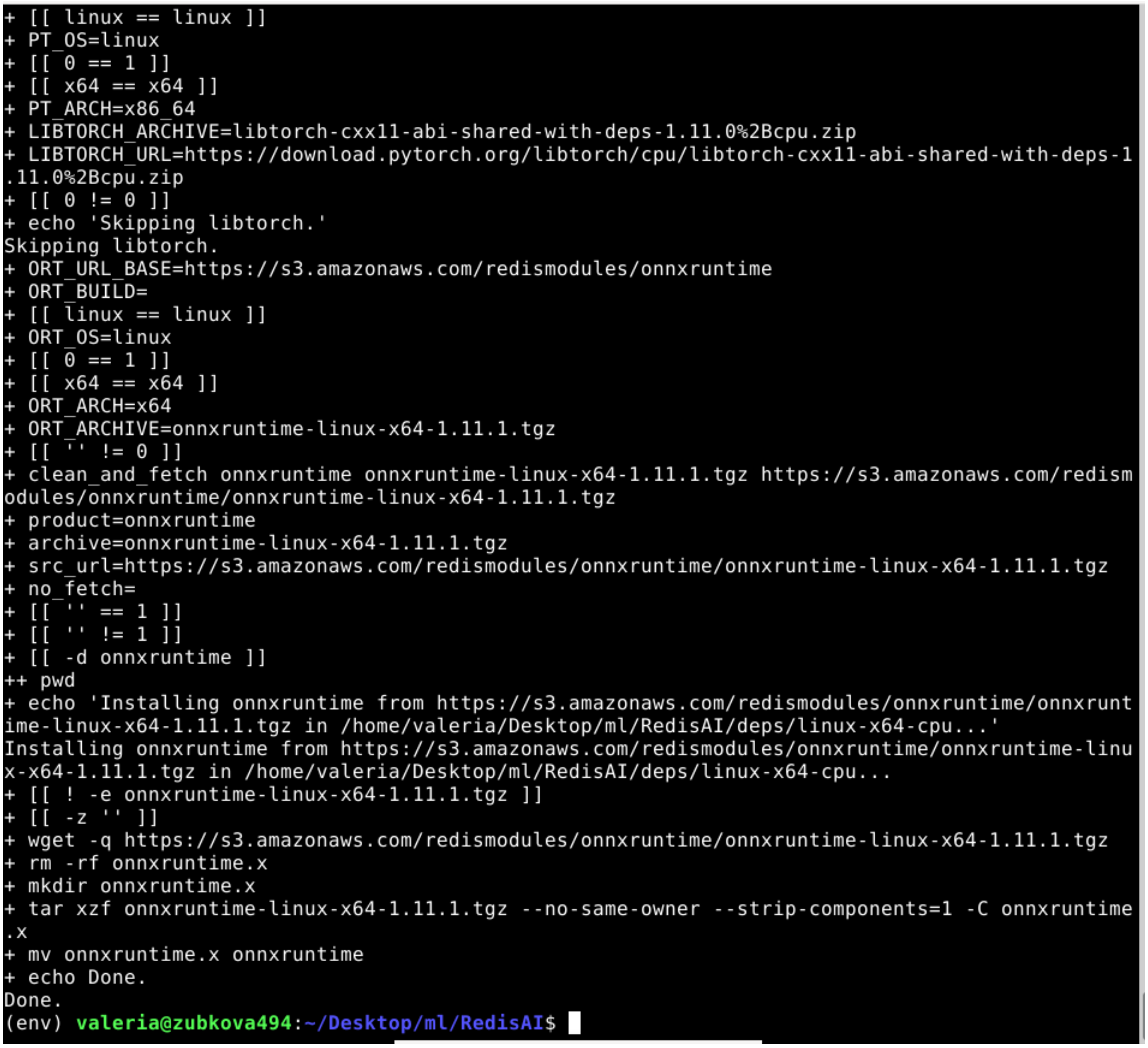
```
    CC listpack.o
    CC localtime.o
    CC lolwut.o
    CC lolwut5.o
    CC lolwut6.o
    CC acl.o
    CC tracking.o
    CC socket.o
    CC tls.o
    CC sha256.o
    CC timeout.o
    CC setcpuaffinity.o
    CC monotonic.o
    CC mt19937-64.o
    CC resp_parser.o
    CC call_reply.o
    CC script_lua.o
    CC script.o
    CC functions.o
    CC function_lua.o
    GEN commands.c
Processing json files...
Linking container command to subcommands...
Checking all commands...
Generating commands.c...
All done, exiting.
    CC commands.o
    CC strl.o
    CC connection.o
    CC unix.o
    LINK redis-server
    INSTALL redis-sentinel
    CC redis-cli.o
    CC redisassert.o
    CC cli_common.o
    LINK redis-cli
    CC redis-benchmark.o
    LINK redis-benchmark
    INSTALL redis-check-rdb
    INSTALL redis-check-aof

Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory '/home/valeria/Desktop/ml/redis/src'
(env) valeria@zubkova494:~/Desktop/ml/redis$ 
```

Рисунок 4 – Установка Redis



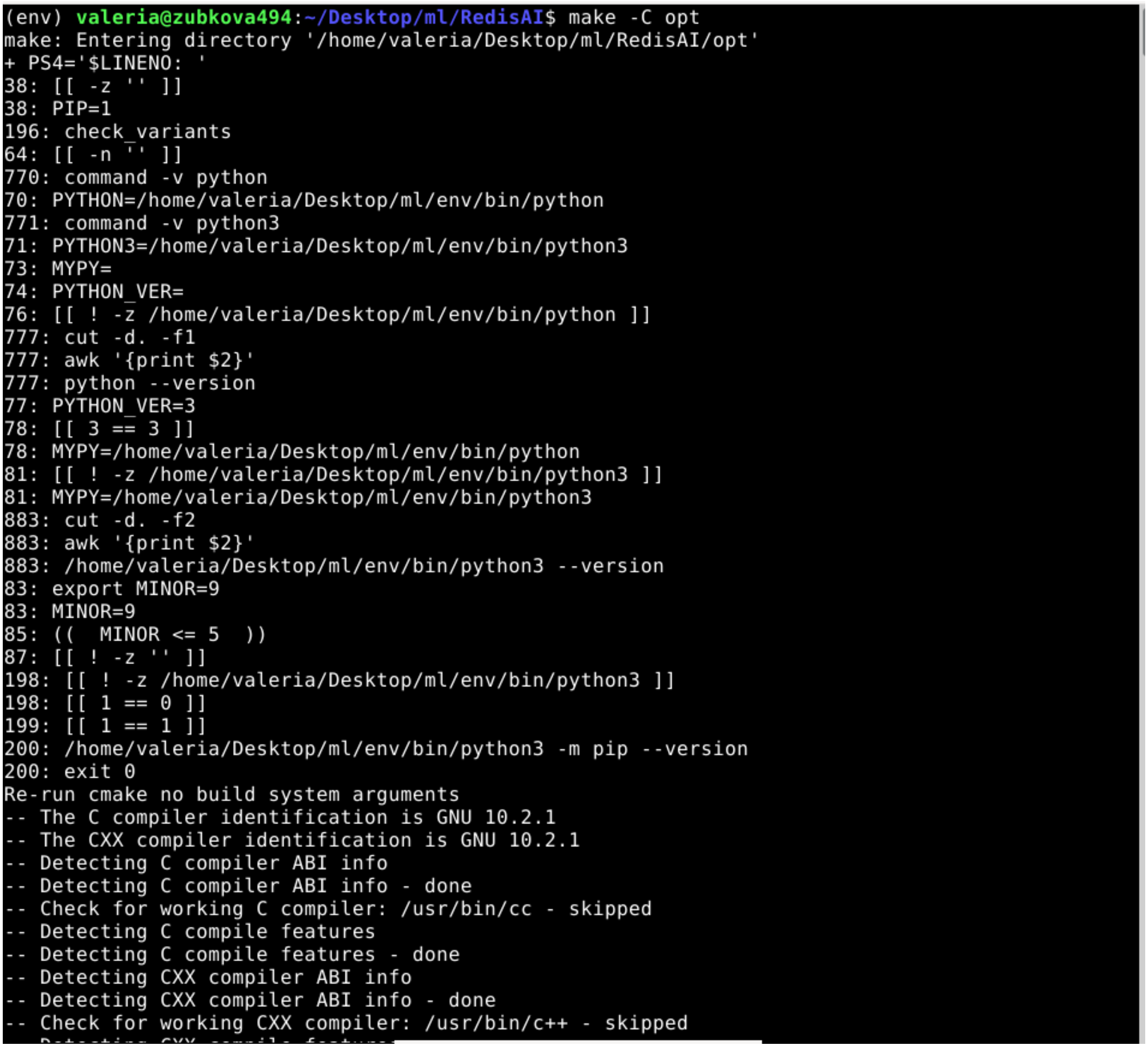
```
make[1]: Leaving directory '/home/valeria/Desktop/ml/redis/src'
(env) valeria@zubkova494:~/Desktop/ml/redis$ cd ..
(env) valeria@zubkova494:~/Desktop/ml$ cd RedisAI
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$ export CPU=1
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$ export VERBOSE=1
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$ export WITH_TF=0
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$ export WITH_TFLITE=0
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$ export WITH_PT=0
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$ bash get_deps.sh cpu
```

Рисунок 5 – Установка RedisAI



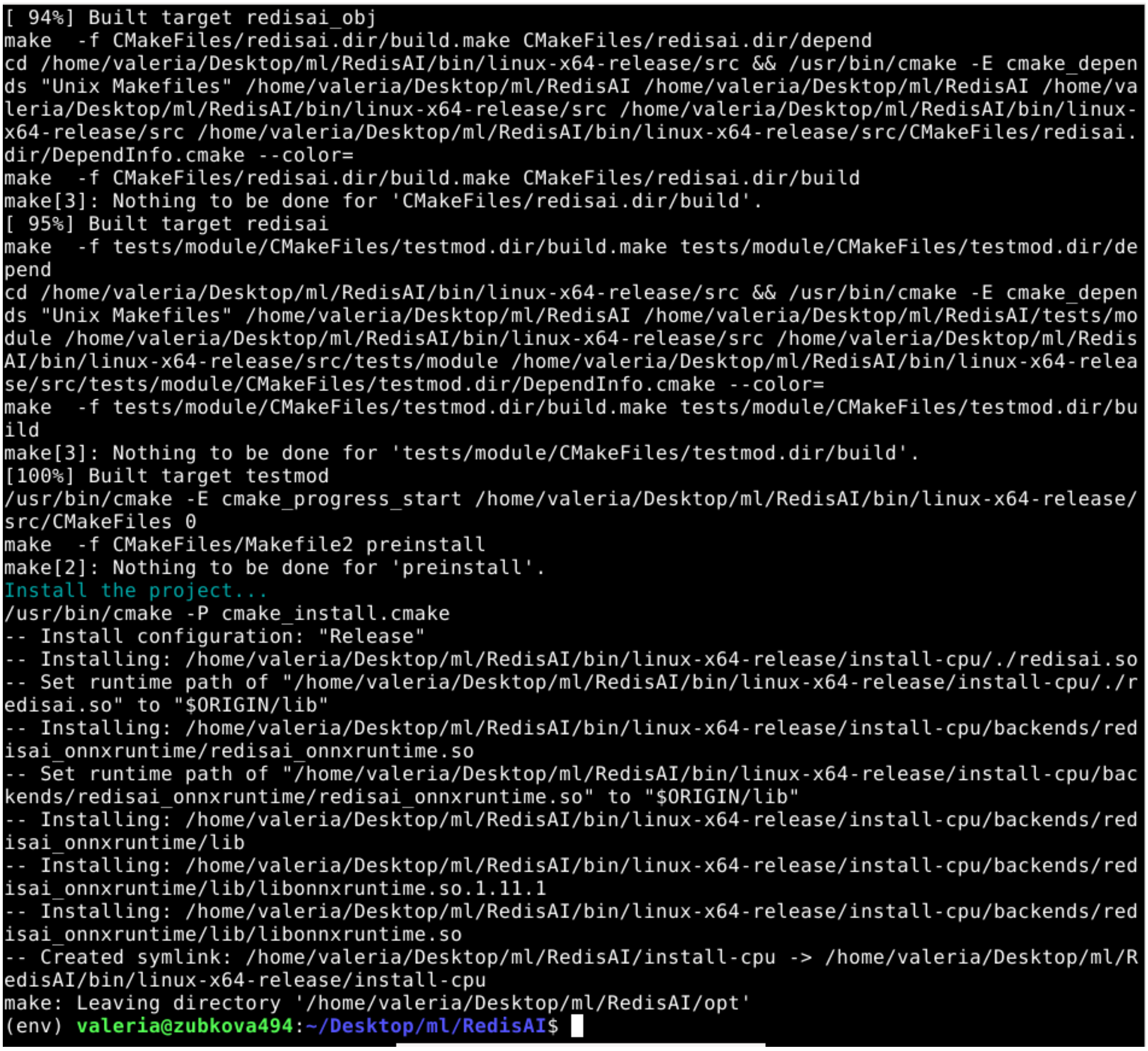
```
+ [[ linux == linux ]]
+ PT_OS=linux
+ [[ 0 == 1 ]]
+ [[ x64 == x64 ]]
+ PT_ARCH=x86_64
+ LIBTORCH_ARCHIVE=libtorch-cxx11-abi-shared-with-deps-1.11.0%2Bcpu.zip
+ LIBTORCH_URL=https://download.pytorch.org/libtorch/cpu/libtorch-cxx11-abi-shared-with-deps-1
.11.0%2Bcpu.zip
+ [[ 0 != 0 ]]
+ echo 'Skipping libtorch.'
Skipping libtorch.
+ ORT_URL_BASE=https://s3.amazonaws.com/redismodules/onnxruntime
+ ORT_BUILD=
+ [[ linux == linux ]]
+ ORT_OS=linux
+ [[ 0 == 1 ]]
+ [[ x64 == x64 ]]
+ ORT_ARCH=x64
+ ORT_ARCHIVE=onnxruntime-linux-x64-1.11.1.tgz
+ [[ '' != 0 ]]
+ clean_and_fetch onnxruntime onnxruntime-linux-x64-1.11.1.tgz https://s3.amazonaws.com/redism
odules/onnxruntime/onnxruntime-linux-x64-1.11.1.tgz
+ product=onnxruntime
+ archive=onnxruntime-linux-x64-1.11.1.tgz
+ src_url=https://s3.amazonaws.com/redismodules/onnxruntime/onnxruntime-linux-x64-1.11.1.tgz
+ no_fetch=
+ [[ '' == 1 ]]
+ [[ '' != 1 ]]
+ [[ -d onnxruntime ]]
++ pwd
+ echo 'Installing onnxruntime from https://s3.amazonaws.com/redismodules/onnxruntime/onnxrunt
ime-linux-x64-1.11.1.tgz in /home/valeria/Desktop/ml/RedisAI/deps/linux-x64-cpu...'
Installing onnxruntime from https://s3.amazonaws.com/redismodules/onnxruntime/onnxruntime-linu
x-x64-1.11.1.tgz in /home/valeria/Desktop/ml/RedisAI/deps/linux-x64-cpu...
+ [[ ! -e onnxruntime-linux-x64-1.11.1.tgz ]]
+ [[ -z '' ]]
+ wget -q https://s3.amazonaws.com/redismodules/onnxruntime/onnxruntime-linux-x64-1.11.1.tgz
+ rm -rf onnxruntime.x
+ mkdir onnxruntime.x
+ tar xzf onnxruntime-linux-x64-1.11.1.tgz --no-same-owner --strip-components=1 -C onnxruntime
.x
+ mv onnxruntime.x onnxruntime
+ echo Done.
Done.
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$
```

Рисунок 6 – Успешное завершение установки RedisAI



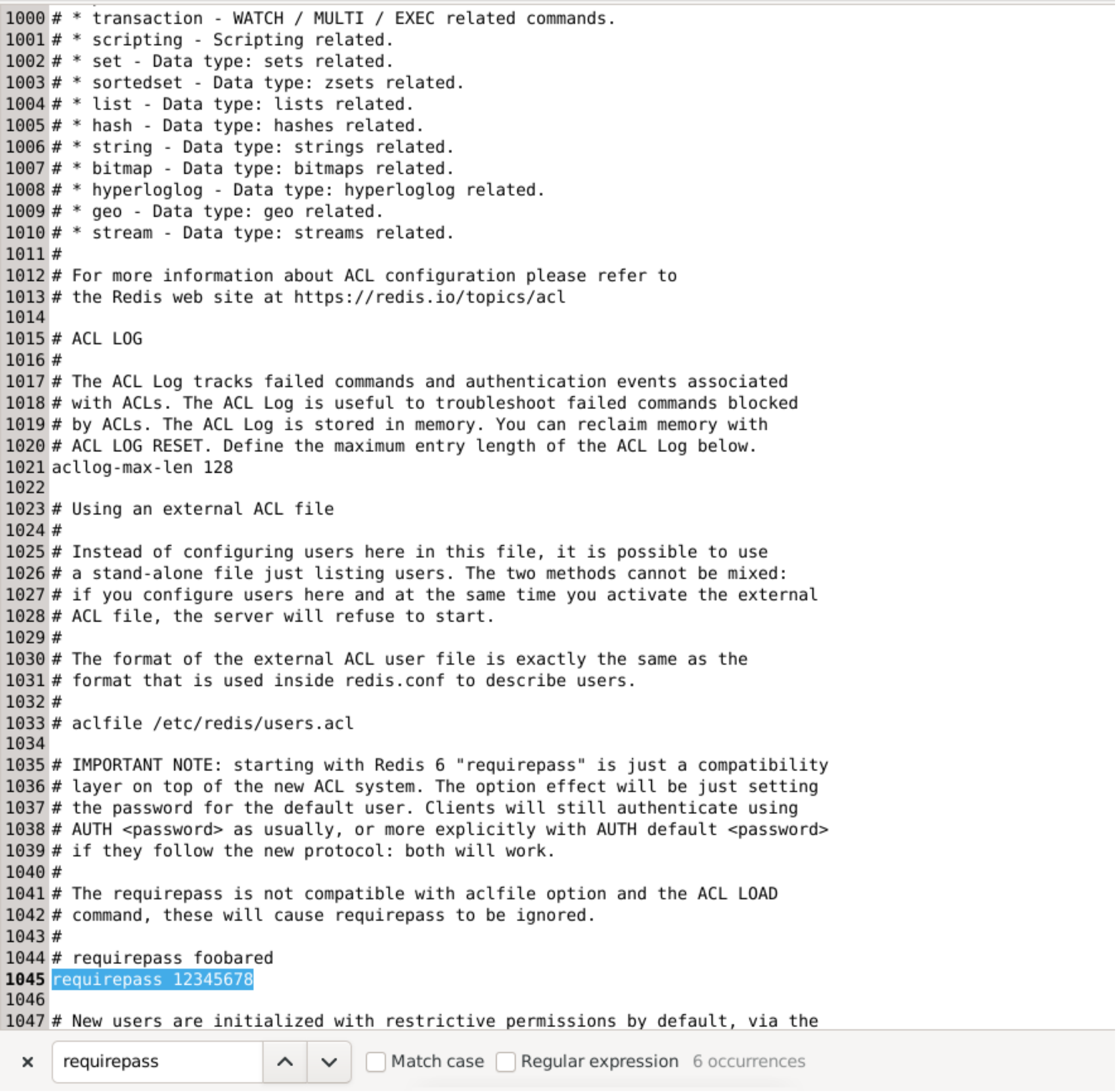
```
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$ make -C opt
make: Entering directory '/home/valeria/Desktop/ml/RedisAI/opt'
+ PS4='$LINENO: '
38: [[ -z '' ]]
38: PIP=1
196: check_variants
64: [[ -n '' ]]
770: command -v python
70: PYTHON=/home/valeria/Desktop/ml/env/bin/python
771: command -v python3
71: PYTHON3=/home/valeria/Desktop/ml/env/bin/python3
73: MYPY=
74: PYTHON_VER=
76: [[ ! -z /home/valeria/Desktop/ml/env/bin/python ]]
777: cut -d. -f1
777: awk '{print $2}'
777: python --version
77: PYTHON_VER=3
78: [[ 3 == 3 ]]
78: MYPY=/home/valeria/Desktop/ml/env/bin/python
81: [[ ! -z /home/valeria/Desktop/ml/env/bin/python3 ]]
81: MYPY=/home/valeria/Desktop/ml/env/bin/python3
883: cut -d. -f2
883: awk '{print $2}'
883: /home/valeria/Desktop/ml/env/bin/python3 --version
83: export MINOR=9
83: MINOR=9
85: ((  MINOR <= 5  ))
87: [[ ! -z '' ]]
198: [[ ! -z /home/valeria/Desktop/ml/env/bin/python3 ]]
198: [[ 1 == 0 ]]
199: [[ 1 == 1 ]]
200: /home/valeria/Desktop/ml/env/bin/python3 -m pip --version
200: exit 0
Re-run cmake no build system arguments
-- The C compiler identification is GNU 10.2.1
-- The CXX compiler identification is GNU 10.2.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
```

Рисунок 7 – Сборка модуля RedisAI



```
[ 94%] Built target redisai_obj
make  -f CMakeFiles/redisai.dir/build.make CMakeFiles/redisai.dir/depend
cd /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/src && /usr/bin/cmake -E cmake_depen
ds "Unix Makefiles" /home/valeria/Desktop/ml/RedisAI /home/valeria/Desktop/ml/RedisAI /home/va
leria/Desktop/ml/RedisAI/bin/linux-x64-release/src /home/valeria/Desktop/ml/RedisAI/bin/linux-
x64-release/src /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/src/CMakeFiles/redisai.
dir/DependInfo.cmake --color=
make  -f CMakeFiles/redisai.dir/build.make CMakeFiles/redisai.dir/build
make[3]: Nothing to be done for 'CMakeFiles/redisai.dir/build'.
[ 95%] Built target redisai
make  -f tests/module/CMakeFiles/testmod.dir/build.make tests/module/CMakeFiles/testmod.dir/de
pend
cd /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/src && /usr/bin/cmake -E cmake_depen
ds "Unix Makefiles" /home/valeria/Desktop/ml/RedisAI /home/valeria/Desktop/ml/RedisAI/tests/mo
dule /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/src /home/valeria/Desktop/ml/Redis
AI/bin/linux-x64-release/src/tests/module /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-relea
se/src/tests/module/CMakeFiles/testmod.dir/DependInfo.cmake --color=
make  -f tests/module/CMakeFiles/testmod.dir/build.make tests/module/CMakeFiles/testmod.dir/bu
ild
make[3]: Nothing to be done for 'tests/module/CMakeFiles/testmod.dir/build'.
[100%] Built target testmod
/usr/bin/cmake -E cmake_progress_start /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/
src/CMakeFiles 0
make  -f CMakeFiles/Makefile2 preinstall
make[2]: Nothing to be done for 'preinstall'.
Install the project...
/usr/bin/cmake -P cmake_install.cmake
-- Install configuration: "Release"
-- Installing: /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/install-cpu/./redisai.so
-- Set runtime path of "/home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/install-cpu/./r
edisai.so" to "$ORIGIN/lib"
-- Installing: /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/install-cpu/backends/red
isai_onnxruntime/redisai_onnxruntime.so
-- Set runtime path of "/home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/install-cpu/bac
kends/redisai_onnxruntime/redisai_onnxruntime.so" to "$ORIGIN/lib"
-- Installing: /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/install-cpu/backends/red
isai_onnxruntime/lib
-- Installing: /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/install-cpu/backends/red
isai_onnxruntime/lib/libonnxruntime.so.1.11.1
-- Installing: /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/install-cpu/backends/red
isai_onnxruntime/lib/libonnxruntime.so
-- Created symlink: /home/valeria/Desktop/ml/RedisAI/install-cpu -> /home/valeria/Desktop/ml/R
edisAI/bin/linux-x64-release/install-cpu
make: Leaving directory '/home/valeria/Desktop/ml/RedisAI/opt'
(env) valeria@zubkova494:~/Desktop/ml/RedisAI$ 
```

Рисунок 8 – Успешное завершение копиляции

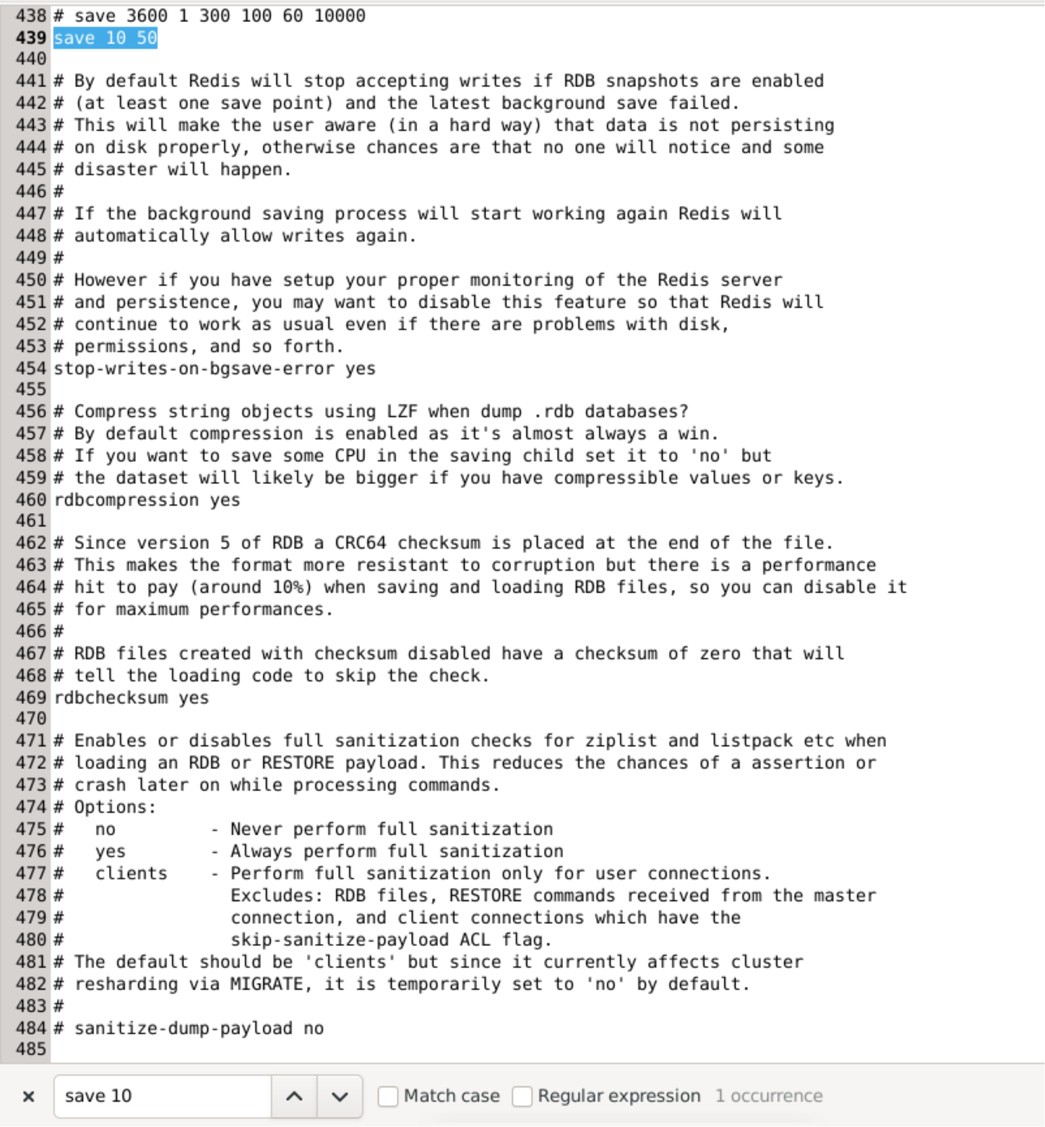


```
# * transaction - WATCH / MULTI / EXEC related commands.
# * scripting - Scripting related.
# * set - Data type: sets related.
# * sortedset - Data type: zsets related.
# * list - Data type: lists related.
# * hash - Data type: hashes related.
# * string - Data type: strings related.
# * bitmap - Data type: bitmaps related.
# * hyperloglog - Data type: hyperloglog related.
# * geo - Data type: geo related.
# * stream - Data type: streams related.
#
# For more information about ACL configuration please refer to
# the Redis web site at https://redis.io/topics/acl

# ACL LOG
#
# The ACL Log tracks failed commands and authentication events associated
# with ACLs. The ACL Log is useful to troubleshoot failed commands blocked
# by ACLs. The ACL Log is stored in memory. You can reclaim memory with
# ACL LOG RESET. Define the maximum entry length of the ACL Log below.
acllog-max-len 128

# Using an external ACL file
#
# Instead of configuring users here in this file, it is possible to use
# a stand-alone file just listing users. The two methods cannot be mixed:
# if you configure users here and at the same time you activate the external
# ACL file, the server will refuse to start.
#
# The format of the external ACL user file is exactly the same as the
# format that is used inside redis.conf to describe users.
#
# aclfile /etc/redis/users.acl

# IMPORTANT NOTE: starting with Redis 6 "requirepass" is just a compatibility
# layer on top of the new ACL system. The option effect will be just setting
# the password for the default user. Clients will still authenticate using
# AUTH <password> as usually, or more explicitly with AUTH default <password>
# if they follow the new protocol: both will work.
#
# The requirepass is not compatible with aclfile option and the ACL LOAD
# command, these will cause requirepass to be ignored.
#
# requirepass foobared
requirepass 12345678

# New users are initialized with restrictive permissions by default, via the
```

requirepass

□ Match case □ Regular expression 6 occurrences

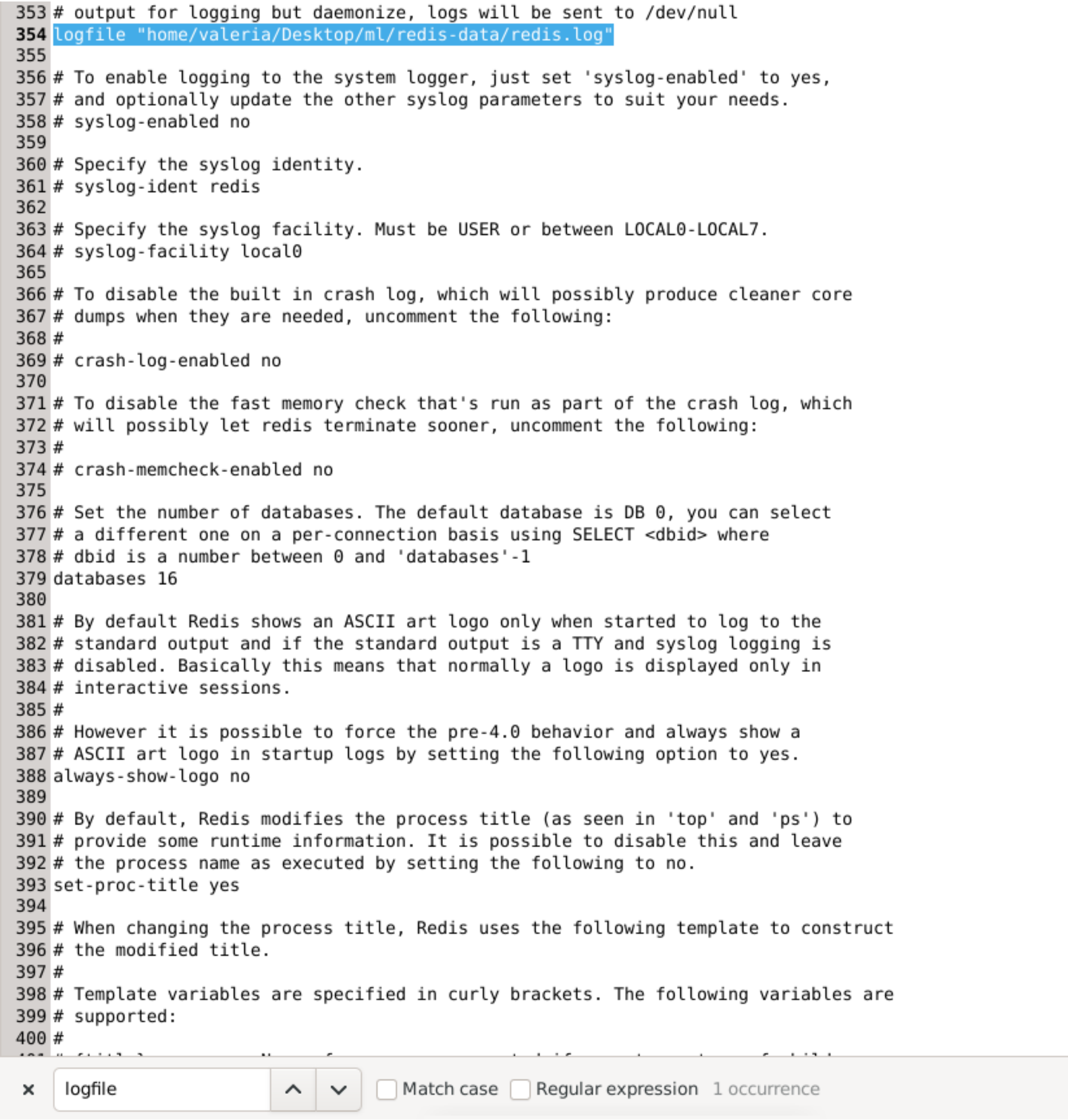
Рисунок 9 – Редактирование requirements.txt



```
# save 3600 1 300 100 60 10000
save 10 50

# By default Redis will stop accepting writes if RDB snapshots are enabled
# (at least one save point) and the latest background save failed.
# This will make the user aware (in a hard way) that data is not persisting
# on disk properly, otherwise chances are that no one will notice and some
# disaster will happen.
#
# If the background saving process will start working again Redis will
# automatically allow writes again.
#
# However if you have setup your proper monitoring of the Redis server
# and persistence, you may want to disable this feature so that Redis will
# continue to work as usual even if there are problems with disk,
# permissions, and so forth.
stop-writes-on-bgsave-error yes

# Compress string objects using LZF when dump .rdb databases?
# By default compression is enabled as it's almost always a win.
# If you want to save some CPU in the saving child set it to 'no' but
# the dataset will likely be bigger if you have compressible values or keys.
rdbcompression yes

# Since version 5 of RDB a CRC64 checksum is placed at the end of the file.
# This makes the format more resistant to corruption but there is a performance
# hit to pay (around 10%) when saving and loading RDB files, so you can disable it
# for maximum performances.
#
# RDB files created with checksum disabled have a checksum of zero that will
# tell the loading code to skip the check.
rdbchecksum yes

# Enables or disables full sanitization checks for ziplist and listpack etc when
# loading an RDB or RESTORE payload. This reduces the chances of a assertion or
# crash later on while processing commands.
# Options:
#   no         - Never perform full sanitization
#   yes        - Always perform full sanitization
#   clients    - Perform full sanitization only for user connections.
#                Excludes: RDB files, RESTORE commands received from the master
#                connection, and client connections which have the
#                skip-sanitize-payload ACL flag.
# The default should be 'clients' but since it currently affects cluster
# resharding via MIGRATE, it is temporarily set to 'no' by default.
#
# sanitize-dump-payload no

```

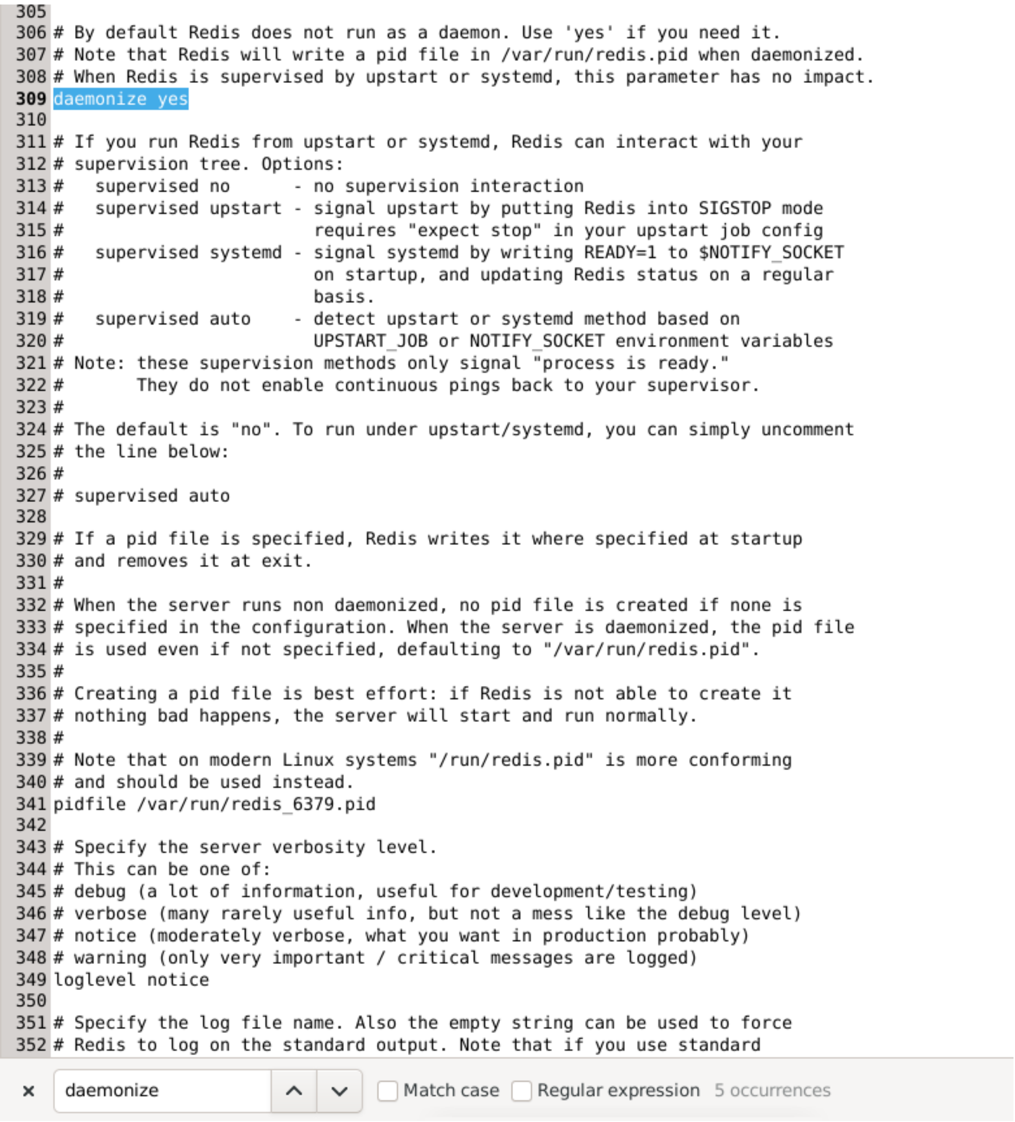
× save 10 ^ ∨ □ Match case □ Regular expression 1 occurrence

Рисунок 10 - Редактирование requirements.txt



```
# output for logging but daemonize, logs will be sent to /dev/null
logfile "home/valeria/Desktop/ml/redis-data/redis.log"

# To enable logging to the system logger, just set 'syslog-enabled' to yes,
# and optionally update the other syslog parameters to suit your needs.
# syslog-enabled no

# Specify the syslog identity.
# syslog-ident redis

# Specify the syslog facility. Must be USER or between LOCAL0-LOCAL7.
# syslog-facility local0

# To disable the built in crash log, which will possibly produce cleaner core
# dumps when they are needed, uncomment the following:
#
# crash-log-enabled no

# To disable the fast memory check that's run as part of the crash log, which
# will possibly let redis terminate sooner, uncomment the following:
#
# crash-memcheck-enabled no

# Set the number of databases. The default database is DB 0, you can select
# a different one on a per-connection basis using SELECT <dbid> where
# dbid is a number between 0 and 'databases'-1
databases 16

# By default Redis shows an ASCII art logo only when started to log to the
# standard output and if the standard output is a TTY and syslog logging is
# disabled. Basically this means that normally a logo is displayed only in
# interactive sessions.
#
# However it is possible to force the pre-4.0 behavior and always show a
# ASCII art logo in startup logs by setting the following option to yes.
always-show-logo no

# By default, Redis modifies the process title (as seen in 'top' and 'ps') to
# provide some runtime information. It is possible to disable this and leave
# the process name as executed by setting the following to no.
set-proc-title yes

# When changing the process title, Redis uses the following template to construct
# the modified title.
#
# Template variables are specified in curly brackets. The following variables are
# supported:
#
```

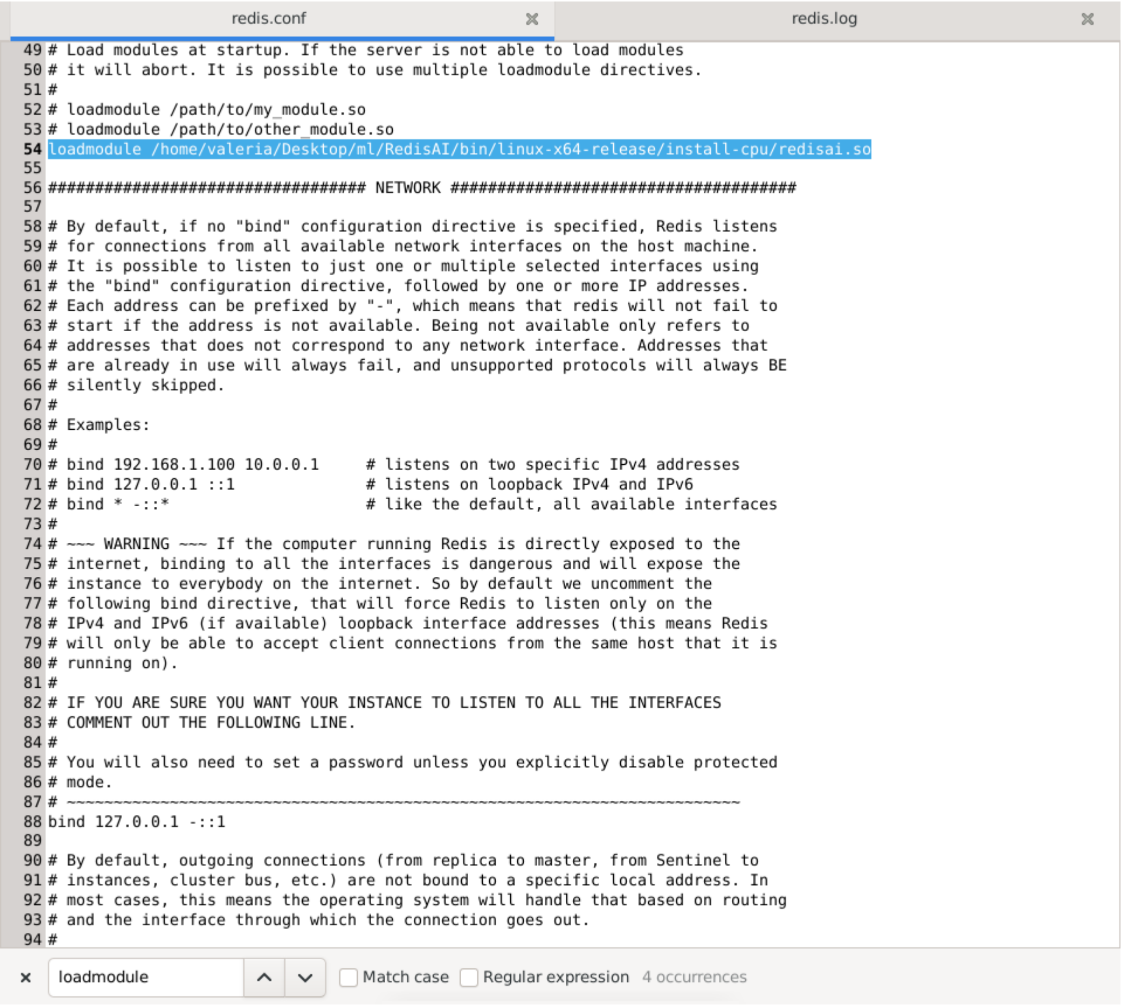
Рисунок 11 - Редактирование requirements.txt



```

# By default Redis does not run as a daemon. Use 'yes' if you need it.
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.
# When Redis is supervised by upstart or systemd, this parameter has no impact.
daemonize yes

# If you run Redis from upstart or systemd, Redis can interact with your
# supervision tree. Options:
#   supervised no      - no supervision interaction
#   supervised upstart - signal upstart by putting Redis into SIGSTOP mode
#                        requires "expect stop" in your upstart job config
#   supervised systemd - signal systemd by writing READY=1 to $NOTIFY_SOCKET
#                        on startup, and updating Redis status on a regular
#                        basis.
#   supervised auto    - detect upstart or systemd method based on
#                        UPSTART_JOB or NOTIFY_SOCKET environment variables
# Note: these supervision methods only signal "process is ready."
#       They do not enable continuous pings back to your supervisor.
#
# The default is "no". To run under upstart/systemd, you can simply uncomment
# the line below:
#
# supervised auto

# If a pid file is specified, Redis writes it where specified at startup
# and removes it at exit.
#
# When the server runs non daemonized, no pid file is created if none is
# specified in the configuration. When the server is daemonized, the pid file
# is used even if not specified, defaulting to "/var/run/redis.pid".
#
# Creating a pid file is best effort: if Redis is not able to create it
# nothing bad happens, the server will start and run normally.
#
# Note that on modern Linux systems "/run/redis.pid" is more conforming
# and should be used instead.
pidfile /var/run/redis_6379.pid

# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel notice

# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
```

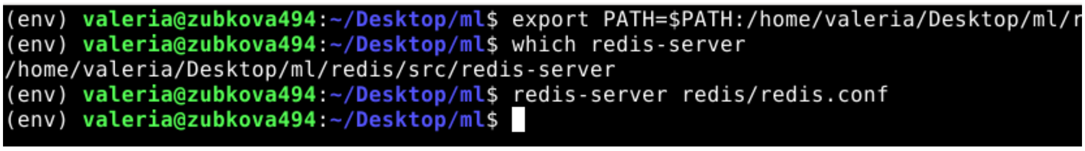
daemonize ☐ Match case ☐ Regular expression 5 occurrences

Рисунок 12 - Редактирование requirements.txt



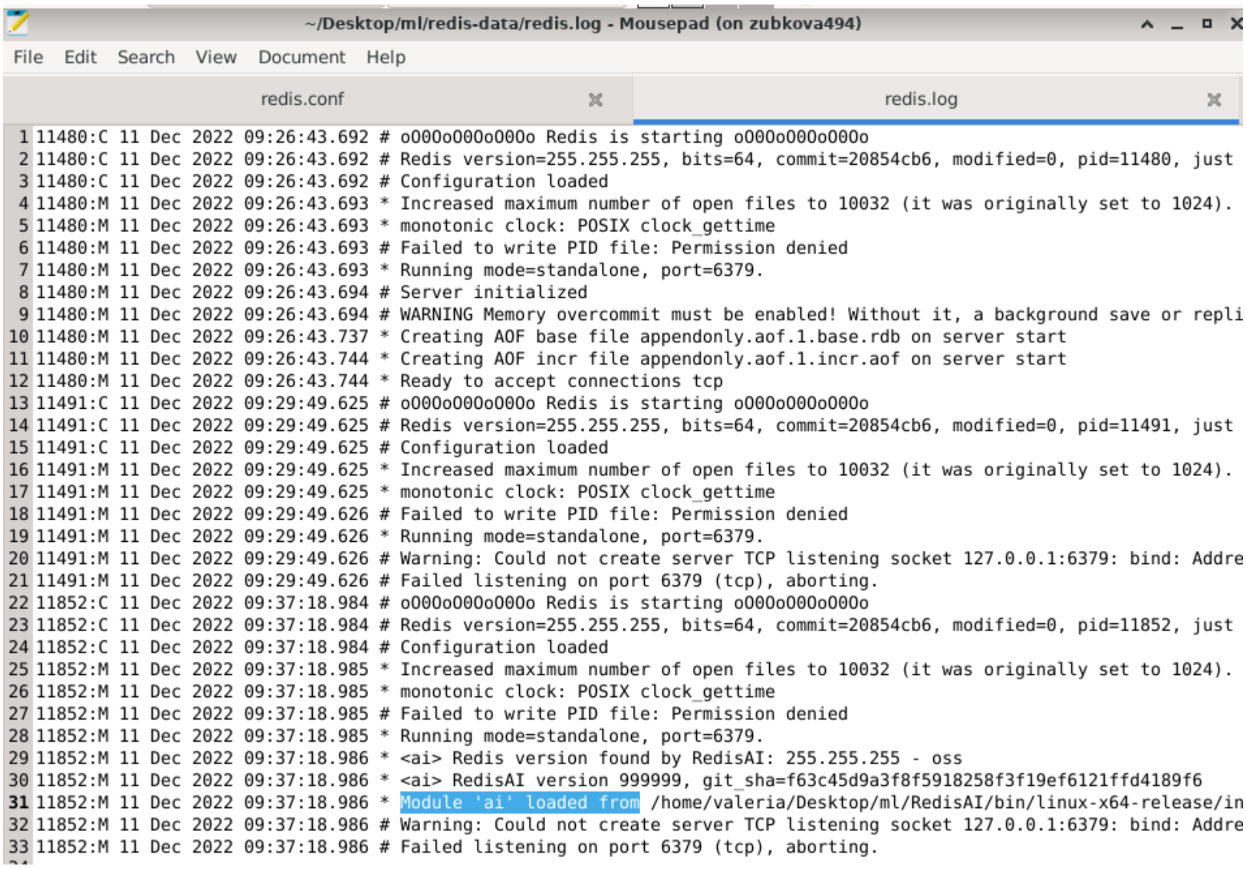
```
# Load modules at startup. If the server is not able to load modules
# it will abort. It is possible to use multiple loadmodule directives.
#
# loadmodule /path/to/my_module.so
# loadmodule /path/to/other_module.so
loadmodule /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/install-cpu/redisai.so

################################## NETWORK #####################################

# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all available network interfaces on the host machine.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
# Each address can be prefixed by "-", which means that redis will not fail to
# start if the address is not available. Being not available only refers to
# addresses that does not correspond to any network interface. Addresses that
# are already in use will always fail, and unsupported protocols will always BE
# silently skipped.
#
# Examples:
#
# bind 192.168.1.100 10.0.0.1     # listens on two specific IPv4 addresses
# bind 127.0.0.1 ::1              # listens on loopback IPv4 and IPv6
# bind * -::*                     # like the default, all available interfaces
#
# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only on the
# IPv4 and IPv6 (if available) loopback interface addresses (this means Redis
# will only be able to accept client connections from the same host that it is
# running on).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# COMMENT OUT THE FOLLOWING LINE.
#
# You will also need to set a password unless you explicitly disable protected
# mode.
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
bind 127.0.0.1 -::1

# By default, outgoing connections (from replica to master, from Sentinel to
# instances, cluster bus, etc.) are not bound to a specific local address. In
# most cases, this means the operating system will handle that based on routing
# and the interface through which the connection goes out.
#
```

Рисунок 13 - Редактирование requirements.txt



```
(env) valeria@zubkova494:~/Desktop/ml$ export PATH=$PATH:/home/valeria/Desktop/ml/r
(env) valeria@zubkova494:~/Desktop/ml$ which redis-server
/home/valeria/Desktop/ml/redis/src/redis-server
(env) valeria@zubkova494:~/Desktop/ml$ redis-server redis/redis.conf
(env) valeria@zubkova494:~/Desktop/ml$ 
```

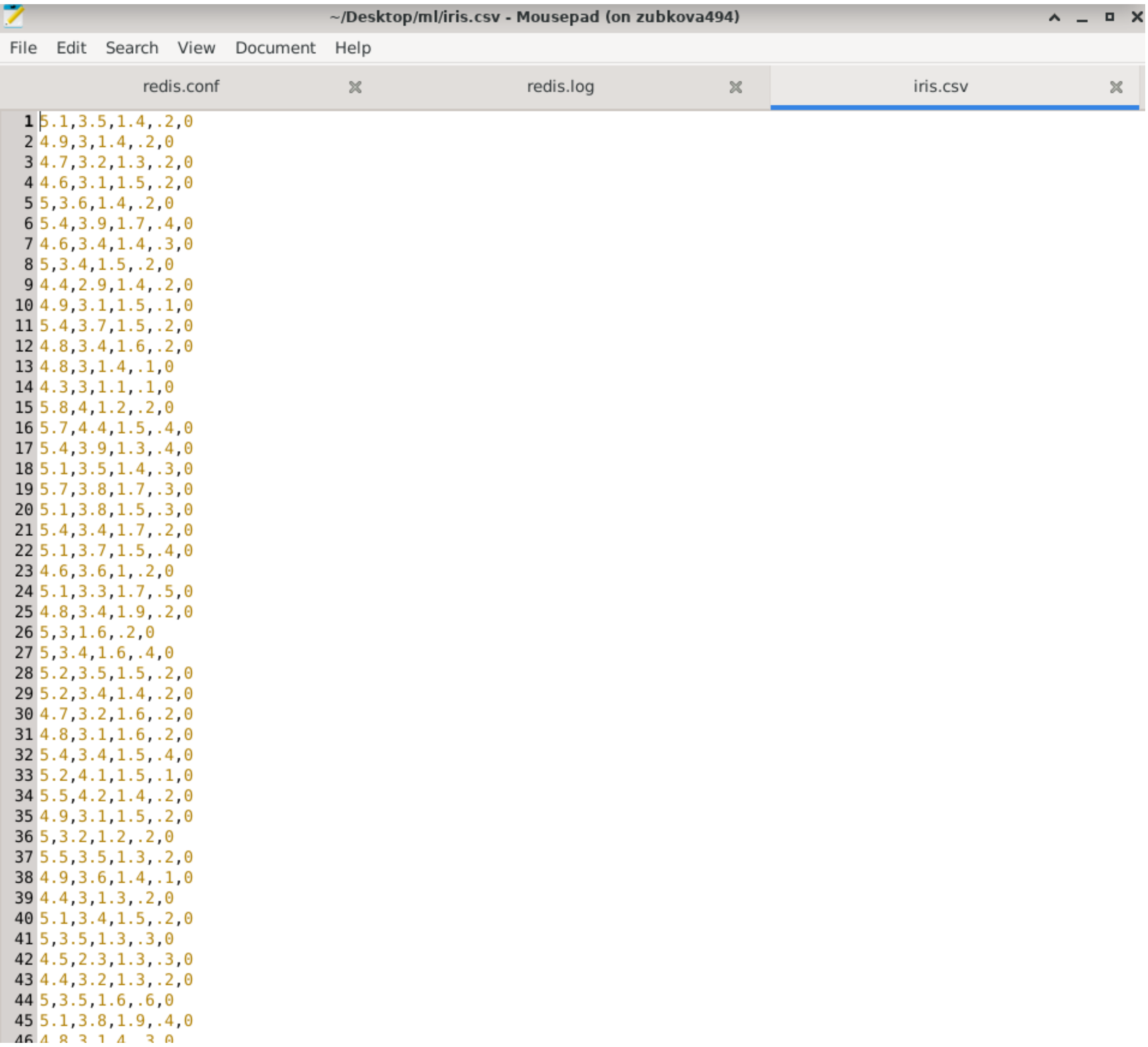
Рисунок 14 - Запуск redis с новой конфигурацией



~/Desktop/ml/redis-data/redis.log - Mousepad (on zubkova494)
File
Edit
Search
View
Document
Help
redis.conf
redis.log

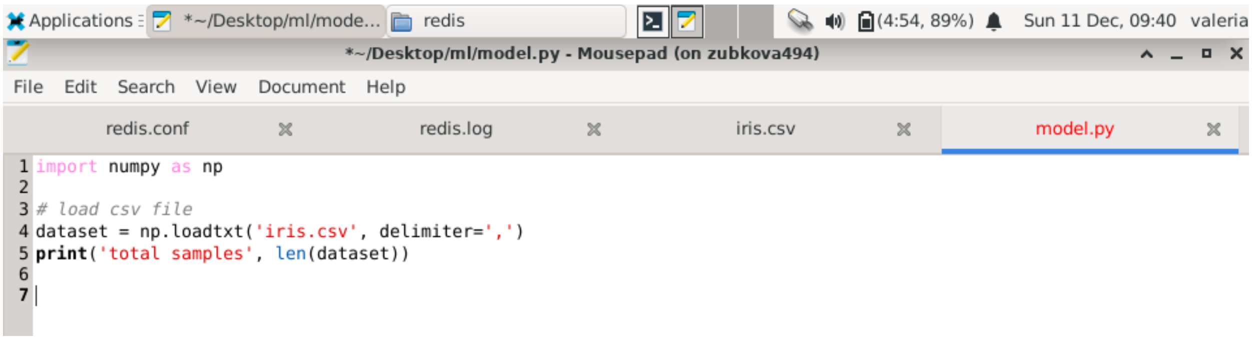
```
11480:C 11 Dec 2022 09:26:43.692 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
11480:C 11 Dec 2022 09:26:43.692 # Redis version=255.255.255, bits=64, commit=20854cb6, modified=0, pid=11480, just
11480:C 11 Dec 2022 09:26:43.692 # Configuration loaded
11480:M 11 Dec 2022 09:26:43.693 * Increased maximum number of open files to 10032 (it was originally set to 1024).
11480:M 11 Dec 2022 09:26:43.693 * monotonic clock: POSIX clock_gettime
11480:M 11 Dec 2022 09:26:43.693 # Failed to write PID file: Permission denied
11480:M 11 Dec 2022 09:26:43.693 * Running mode=standalone, port=6379.
11480:M 11 Dec 2022 09:26:43.694 # Server initialized
11480:M 11 Dec 2022 09:26:43.694 # WARNING Memory overcommit must be enabled! Without it, a background save or repli
11480:M 11 Dec 2022 09:26:43.737 * Creating AOF base file appendonly.aof.1.base.rdb on server start
11480:M 11 Dec 2022 09:26:43.744 * Creating AOF incr file appendonly.aof.1.incr.aof on server start
11480:M 11 Dec 2022 09:26:43.744 * Ready to accept connections tcp
11491:C 11 Dec 2022 09:29:49.625 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
11491:C 11 Dec 2022 09:29:49.625 # Redis version=255.255.255, bits=64, commit=20854cb6, modified=0, pid=11491, just
11491:C 11 Dec 2022 09:29:49.625 # Configuration loaded
11491:M 11 Dec 2022 09:29:49.625 * Increased maximum number of open files to 10032 (it was originally set to 1024).
11491:M 11 Dec 2022 09:29:49.625 * monotonic clock: POSIX clock_gettime
11491:M 11 Dec 2022 09:29:49.626 # Failed to write PID file: Permission denied
11491:M 11 Dec 2022 09:29:49.626 * Running mode=standalone, port=6379.
11491:M 11 Dec 2022 09:29:49.626 # Warning: Could not create server TCP listening socket 127.0.0.1:6379: bind: Addre
11491:M 11 Dec 2022 09:29:49.626 # Failed listening on port 6379 (tcp), aborting.
11852:C 11 Dec 2022 09:37:18.984 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
11852:C 11 Dec 2022 09:37:18.984 # Redis version=255.255.255, bits=64, commit=20854cb6, modified=0, pid=11852, just
11852:C 11 Dec 2022 09:37:18.984 # Configuration loaded
11852:M 11 Dec 2022 09:37:18.985 * Increased maximum number of open files to 10032 (it was originally set to 1024).
11852:M 11 Dec 2022 09:37:18.985 * monotonic clock: POSIX clock_gettime
11852:M 11 Dec 2022 09:37:18.985 # Failed to write PID file: Permission denied
11852:M 11 Dec 2022 09:37:18.985 * Running mode=standalone, port=6379.
11852:M 11 Dec 2022 09:37:18.986 * <ai> Redis version found by RedisAI: 255.255.255 - oss
11852:M 11 Dec 2022 09:37:18.986 * <ai> RedisAI version 999999, git_sha=f63c45d9a3f8f5918258f3f19ef6121ffd4189f6
11852:M 11 Dec 2022 09:37:18.986 * Module 'ai' loaded from /home/valeria/Desktop/ml/RedisAI/bin/linux-x64-release/in
11852:M 11 Dec 2022 09:37:18.986 # Warning: Could not create server TCP listening socket 127.0.0.1:6379: bind: Addre
11852:M 11 Dec 2022 09:37:18.986 # Failed listening on port 6379 (tcp), aborting.
```

Рисунок 15 – сообщение о том, что RedisAI подключен


~/Desktop/ml/iris.csv - Mousepad (on zubkova494)
File Edit Search View Document Help
redis.conf
redis.log
iris.csv

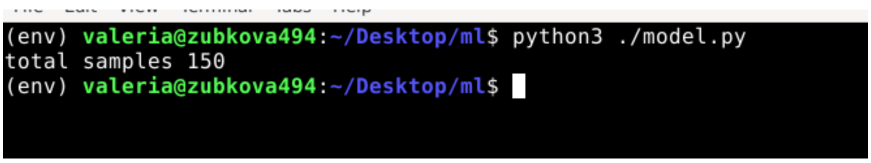
```
5.1,3.5,1.4,.2,0
4.9,3,1.4,.2,0
4.7,3.2,1.3,.2,0
4.6,3.1,1.5,.2,0
5,3.6,1.4,.2,0
5.4,3.9,1.7,.4,0
4.6,3.4,1.4,.3,0
5,3.4,1.5,.2,0
4.4,2.9,1.4,.2,0
4.9,3.1,1.5,.1,0
5.4,3.7,1.5,.2,0
4.8,3.4,1.6,.2,0
4.8,3,1.4,.1,0
4.3,3,1.1,.1,0
5.8,4,1.2,.2,0
5.7,4.4,1.5,.4,0
5.4,3.9,1.3,.4,0
5.1,3.5,1.4,.3,0
5.7,3.8,1.7,.3,0
5.1,3.8,1.5,.3,0
5.4,3.4,1.7,.2,0
5.1,3.7,1.5,.4,0
4.6,3.6,1,.2,0
5.1,3.3,1.7,.5,0
4.8,3.4,1.9,.2,0
5,3,1.6,.2,0
5,3.4,1.6,.4,0
5.2,3.5,1.5,.2,0
5.2,3.4,1.4,.2,0
4.7,3.2,1.6,.2,0
4.8,3.1,1.6,.2,0
5.4,3.4,1.5,.4,0
5.2,4.1,1.5,.1,0
5.5,4.2,1.4,.2,0
4.9,3.1,1.5,.2,0
5,3.2,1.2,.2,0
5.5,3.5,1.3,.2,0
4.9,3.6,1.4,.1,0
4.4,3,1.3,.2,0
5.1,3.4,1.5,.2,0
5,3.5,1.3,.3,0
4.5,2.3,1.3,.3,0
4.4,3.2,1.3,.2,0
5,3.5,1.6,.6,0
5.1,3.8,1.9,.4,0
4.8,3,1.4,.3,0
```

Рисунок 16 - Редактирование csv файла для работы


Applications
*~/Desktop/ml/mode...
redis
(4:54, 89%)
Sun 11 Dec, 09:40 valeria
*~/Desktop/ml/model.py - Mousepad (on zubkova494)
File Edit Search View Document Help
redis.conf
redis.log
iris.csv
model.py

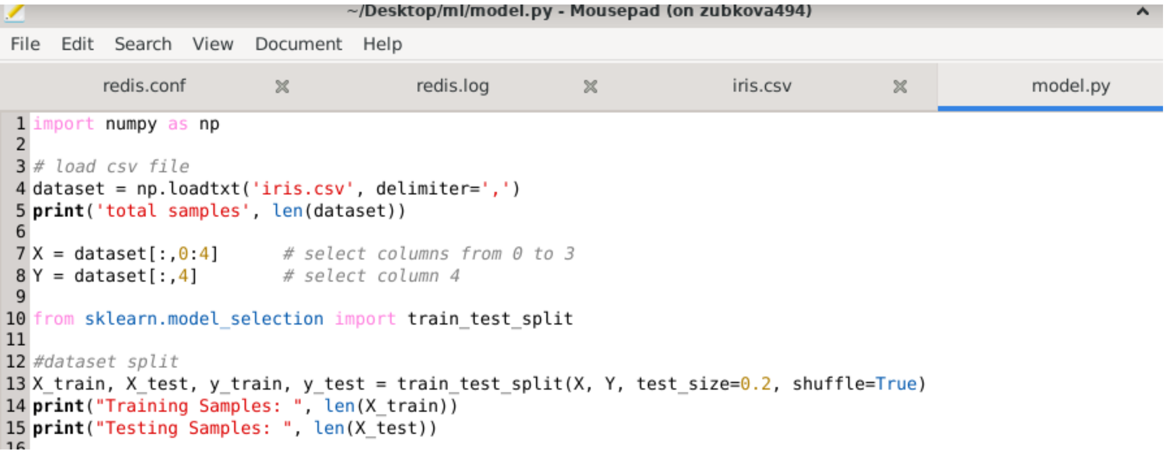
```python
import numpy as np

# load csv file
dataset = np.loadtxt('iris.csv', delimiter=',')
print('total samples', len(dataset))

```

Рисунок 17 - создание model.py



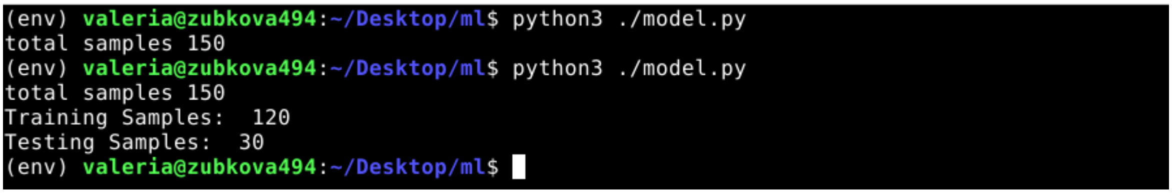
```
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
(env) valeria@zubkova494:~/Desktop/ml$ 
```

Рисунок 18 - запуск model.py


~/Desktop/ml/model.py - Mousepad (on zubkova494)
File Edit Search View Document Help
redis.conf
redis.log
iris.csv
model.py

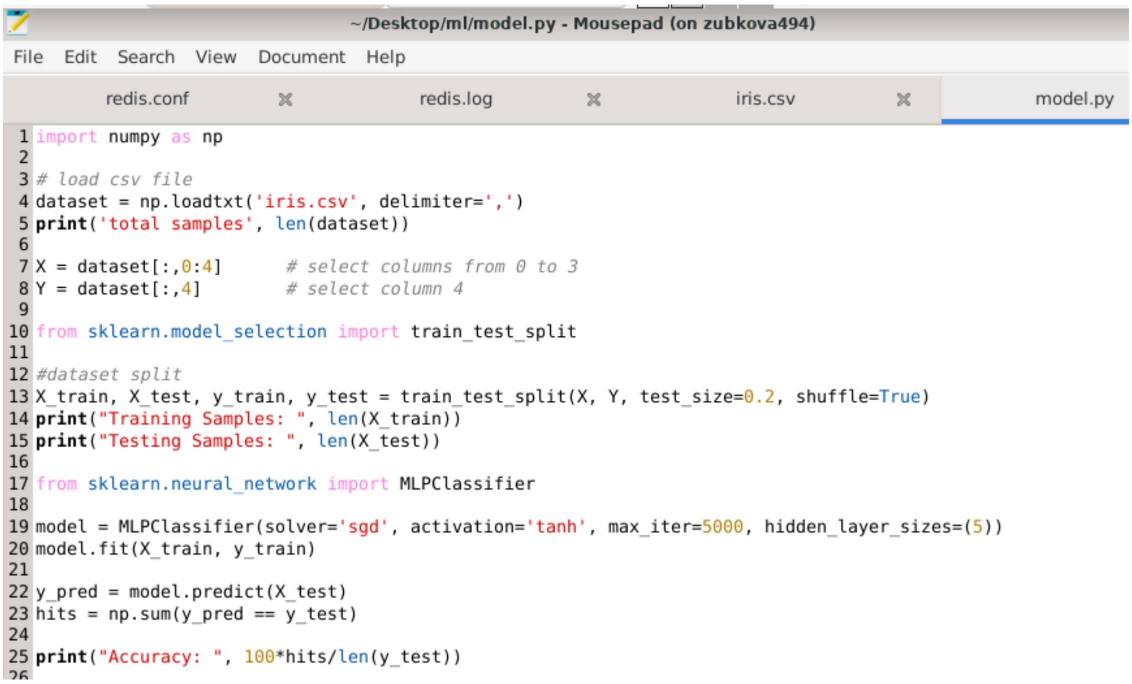
```python
import numpy as np

# load csv file
dataset = np.loadtxt('iris.csv', delimiter=',')
print('total samples', len(dataset))

X = dataset[:,0:4]      # select columns from 0 to 3
Y = dataset[:,4]        # select column 4

from sklearn.model_selection import train_test_split

#dataset split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, shuffle=True)
print("Training Samples: ", len(X_train))
print("Testing Samples: ", len(X_test))
```

Рисунок 19 - Добавление нового функционала в model.py



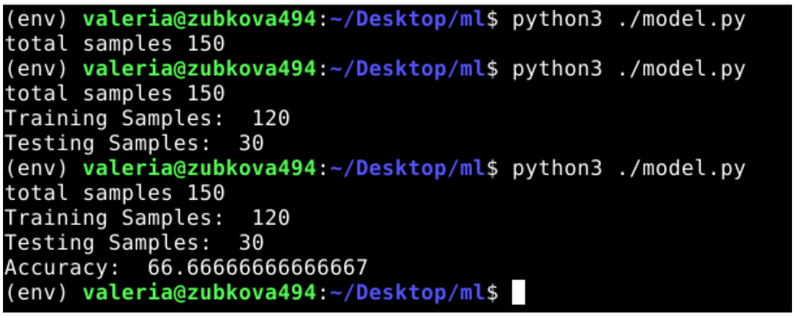
```
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
Training Samples:  120
Testing Samples:  30
(env) valeria@zubkova494:~/Desktop/ml$ 
```

Рисунок 20 - повторный запуск model.py


~/Desktop/ml/model.py - Mousepad (on zubkova494)
File Edit Search View Document Help
redis.conf
redis.log
iris.csv
model.py

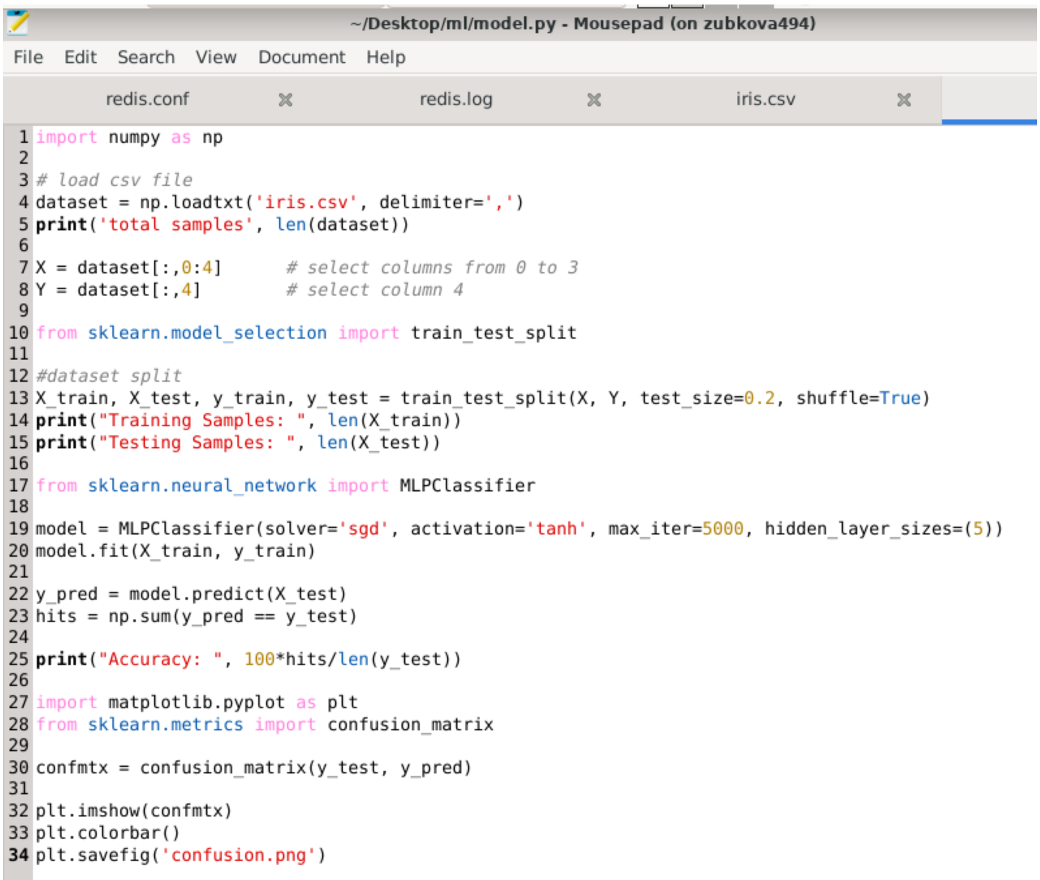
```python
import numpy as np

# load csv file
dataset = np.loadtxt('iris.csv', delimiter=',')
print('total samples', len(dataset))

X = dataset[:,0:4]      # select columns from 0 to 3
Y = dataset[:,4]        # select column 4

from sklearn.model_selection import train_test_split

#dataset split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, shuffle=True)
print("Training Samples: ", len(X_train))
print("Testing Samples: ", len(X_test))

from sklearn.neural_network import MLPClassifier

model = MLPClassifier(solver='sgd', activation='tanh', max_iter=5000, hidden_layer_sizes=(5))
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
hits = np.sum(y_pred == y_test)

print("Accuracy: ", 100*hits/len(y_test))
```

Рисунок 21 - Редактирование model.py



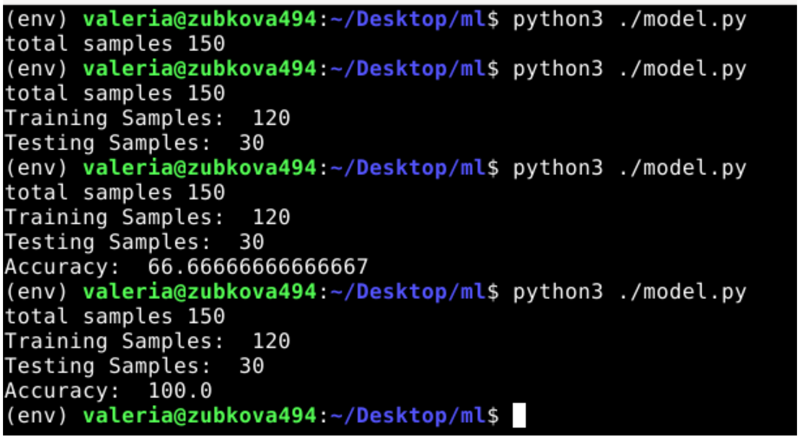
```
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
Training Samples:  120
Testing Samples:  30
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
Training Samples:  120
Testing Samples:  30
Accuracy:  66.66666666666667
(env) valeria@zubkova494:~/Desktop/ml$ 
```

Рисунок 22 - Запуск model.py



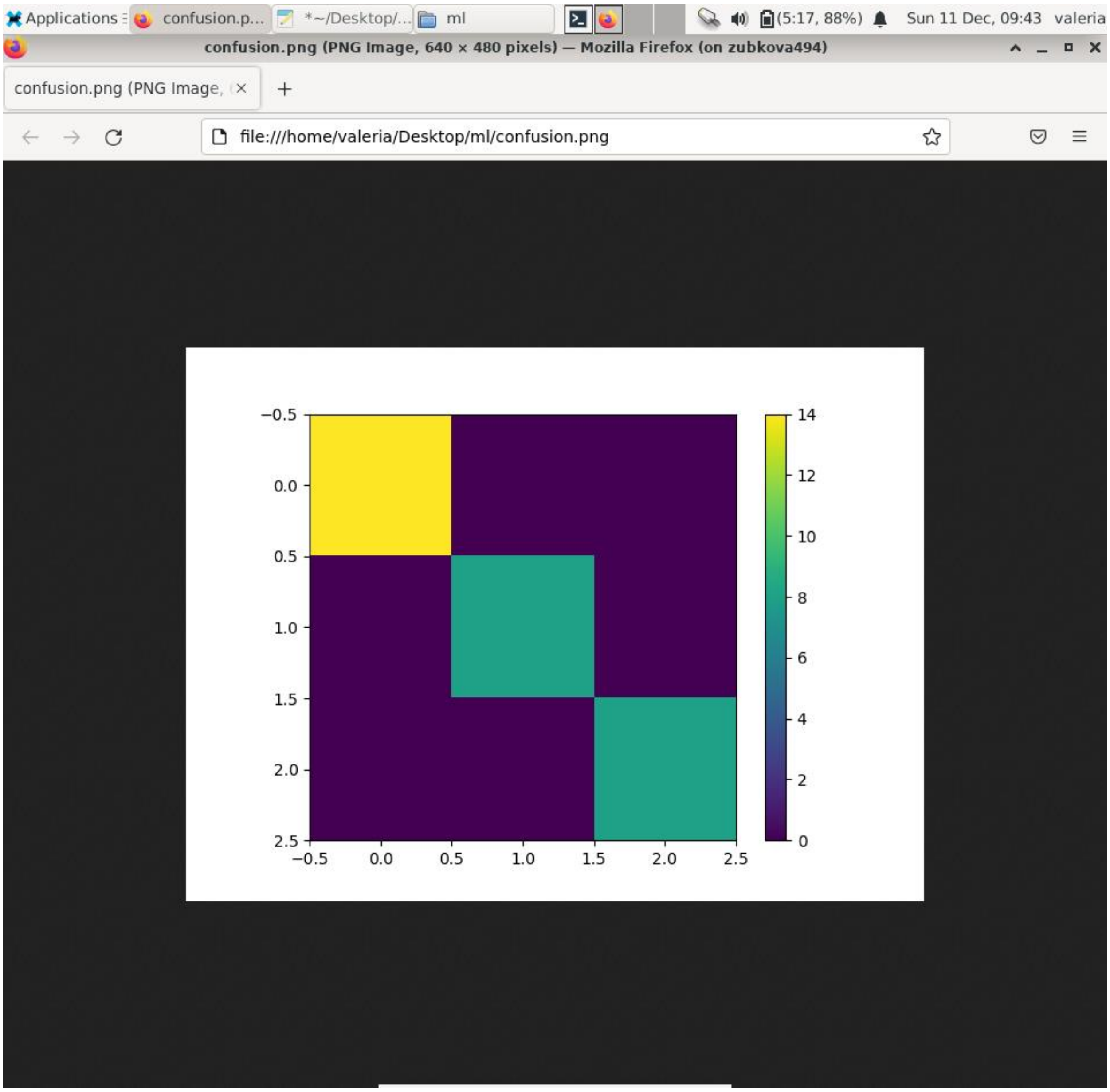
```python
import numpy as np

# load csv file
dataset = np.loadtxt('iris.csv', delimiter=',')
print('total samples', len(dataset))

X = dataset[:,0:4]       # select columns from 0 to 3
Y = dataset[:,4]         # select column 4

from sklearn.model_selection import train_test_split

#dataset split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, shuffle=True)
print("Training Samples: ", len(X_train))
print("Testing Samples: ", len(X_test))

from sklearn.neural_network import MLPClassifier

model = MLPClassifier(solver='sgd', activation='tanh', max_iter=5000, hidden_layer_sizes=(5))
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
hits = np.sum(y_pred == y_test)

print("Accuracy: ", 100*hits/len(y_test))

import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

confmtx = confusion_matrix(y_test, y_pred)

plt.imshow(confmtx)
plt.colorbar()
plt.savefig('confusion.png')
```

Рисунок 23 - Редактирование model.py



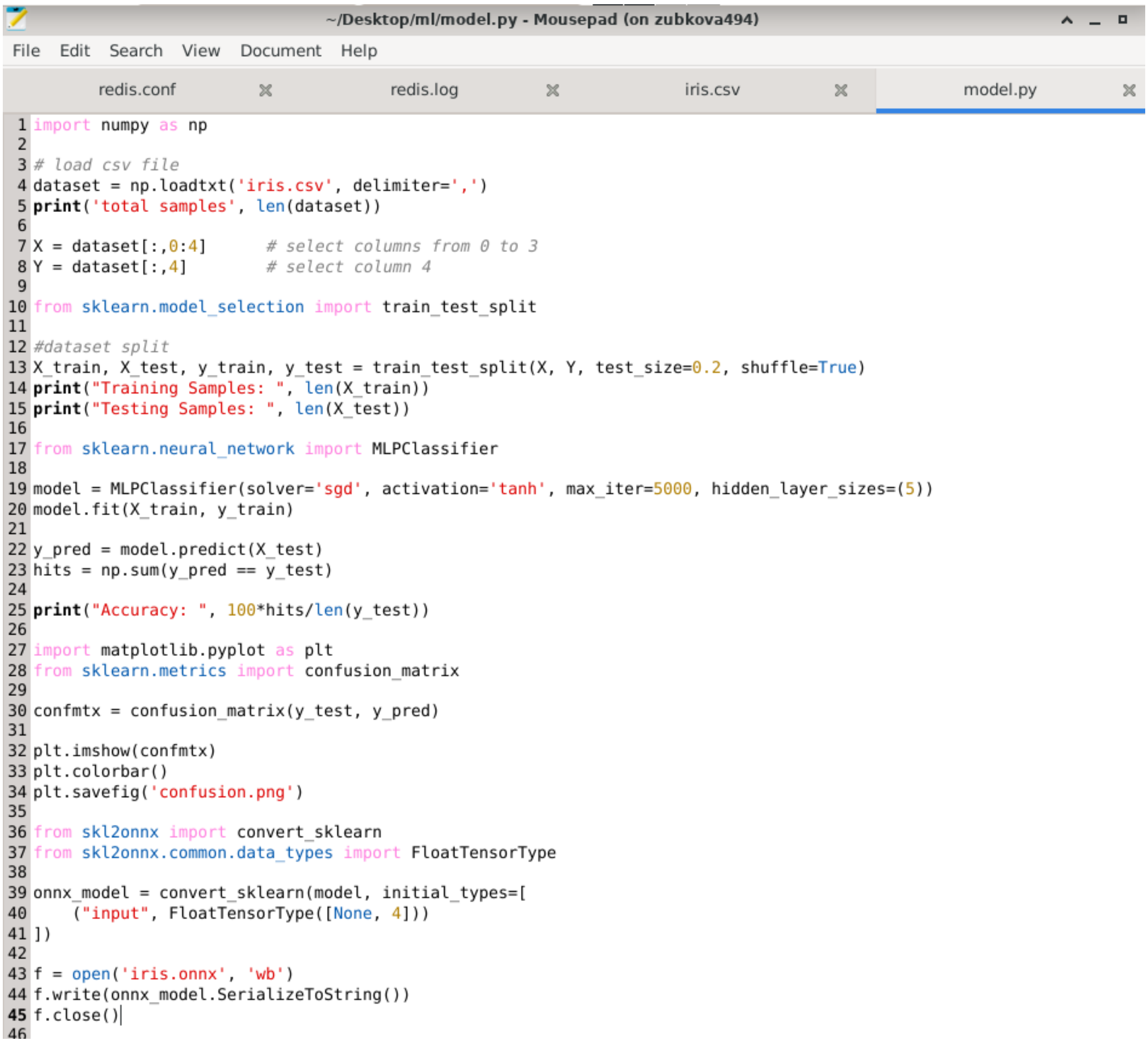
```
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
Training Samples:  120
Testing Samples:  30
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
Training Samples:  120
Testing Samples:  30
Accuracy:  66.66666666666667
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
Training Samples:  120
Testing Samples:  30
Accuracy:  100.0
(env) valeria@zubkova494:~/Desktop/ml$ 
```

Рисунок 24 - Запуск model.py



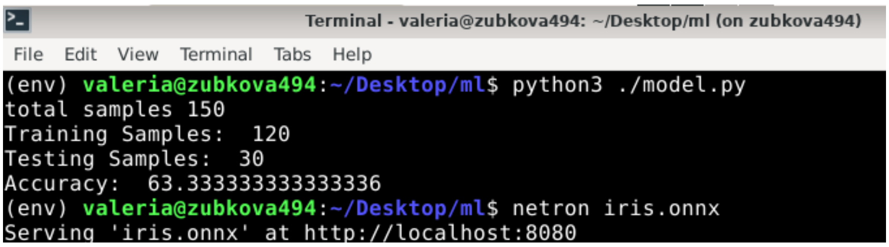
Applications
confusion.p...
*~/Desktop/...
ml
(5:17, 88%)
Sun 11 Dec, 09:43
valeria
confusion.png (PNG Image, 640 × 480 pixels) — Mozilla Firefox (on zubkova494)
confusion.png (PNG Image,
file:///home/valeria/Desktop/ml/confusion.png
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
14
12
10
8
6
4
2
0

Рисунок 25 - Созданная картинка conf.png


~/Desktop/ml/model.py - Mousepad (on zubkova494)
File Edit Search View Document Help
redis.conf
redis.log
iris.csv
model.py

```python
import numpy as np

# load csv file
dataset = np.loadtxt('iris.csv', delimiter=',')
print('total samples', len(dataset))

X = dataset[:,0:4]       # select columns from 0 to 3
Y = dataset[:,4]         # select column 4

from sklearn.model_selection import train_test_split

#dataset split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, shuffle=True)
print("Training Samples: ", len(X_train))
print("Testing Samples: ", len(X_test))

from sklearn.neural_network import MLPClassifier

model = MLPClassifier(solver='sgd', activation='tanh', max_iter=5000, hidden_layer_sizes=(5))
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
hits = np.sum(y_pred == y_test)

print("Accuracy: ", 100*hits/len(y_test))

import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

confmtx = confusion_matrix(y_test, y_pred)

plt.imshow(confmtx)
plt.colorbar()
plt.savefig('confusion.png')

from skl2onnx import convert_sklearn
from skl2onnx.common.data_types import FloatTensorType

onnx_model = convert_sklearn(model, initial_types=[
    ("input", FloatTensorType([None, 4]))
])

f = open('iris.onnx', 'wb')
f.write(onnx_model.SerializeToString())
f.close()

```

Рисунок 26 - Редактирование model.py


Terminal - valeria@zubkova494: ~/Desktop/ml (on zubkova494)
File Edit View Terminal Tabs Help

```
(env) valeria@zubkova494:~/Desktop/ml$ python3 ./model.py
total samples 150
Training Samples:  120
Testing Samples:  30
Accuracy:  63.333333333333336
(env) valeria@zubkova494:~/Desktop/ml$ netron iris.onnx
Serving 'iris.onnx' at http://localhost:8080
```

Рисунок 27 - Запуск созданного скриптом файла iris.onnx через netron.

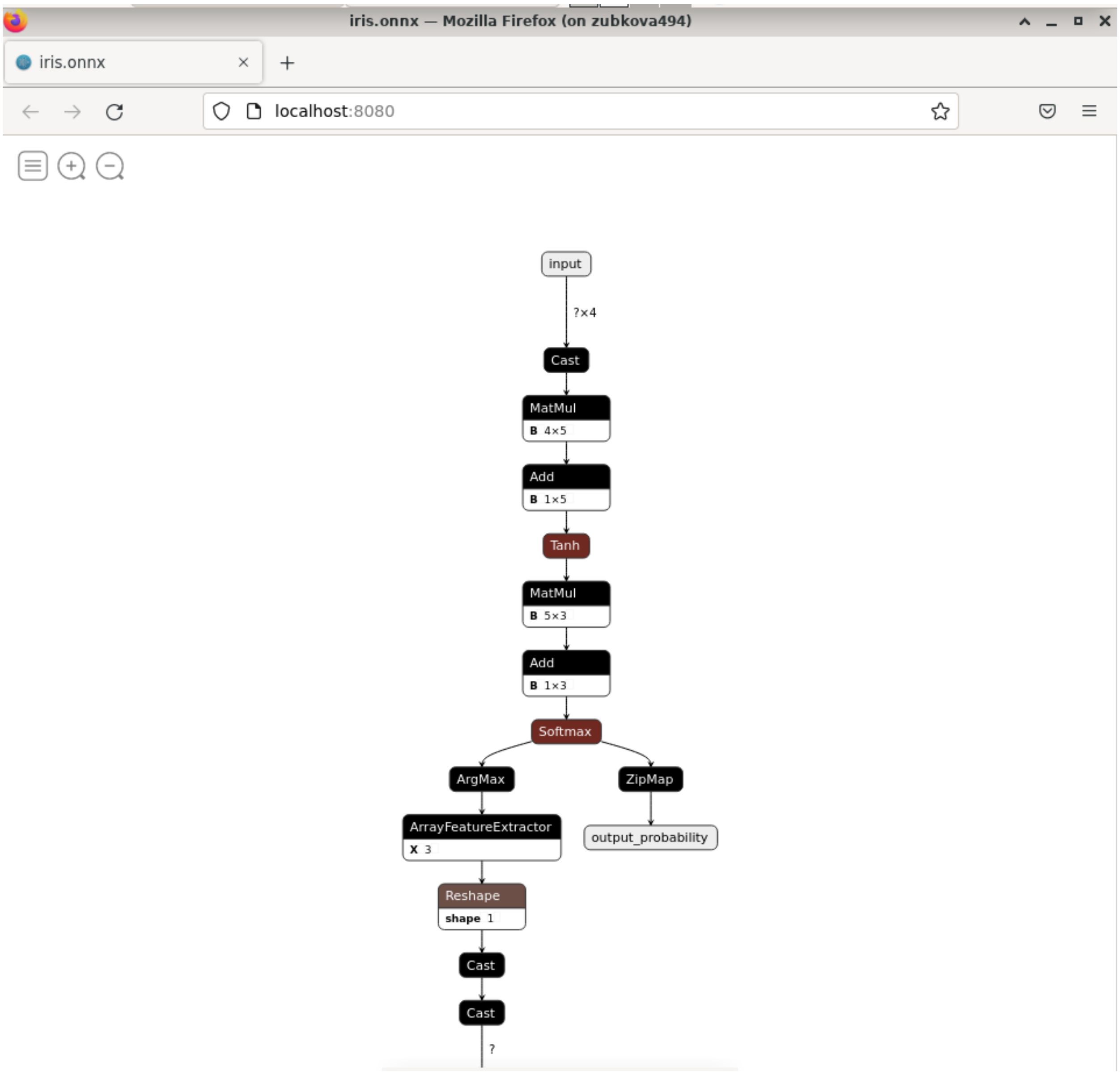
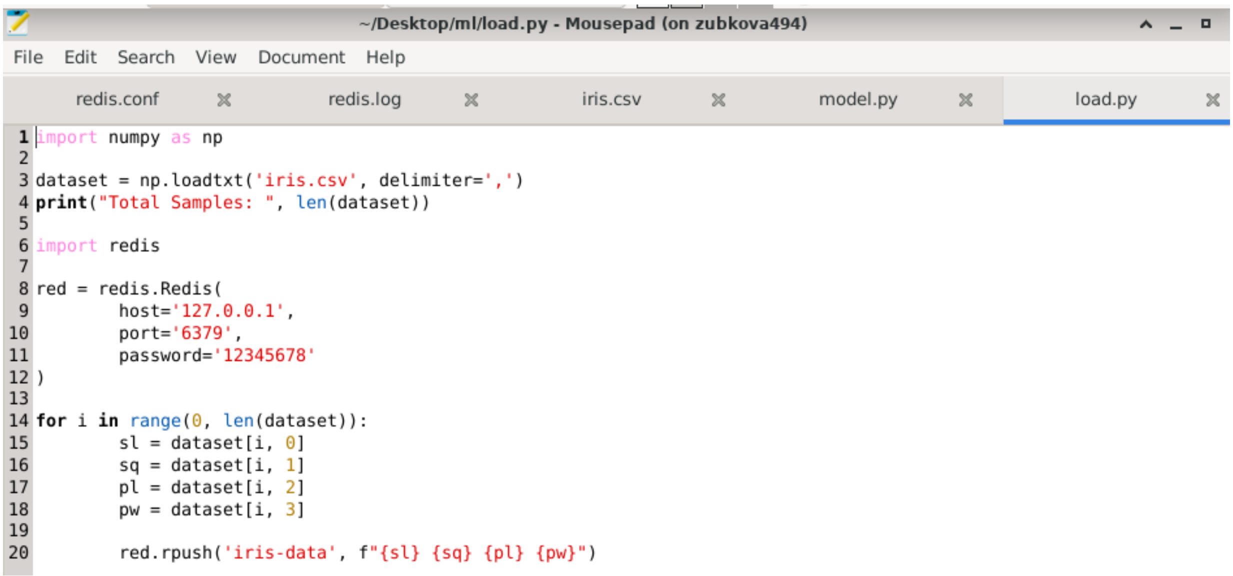
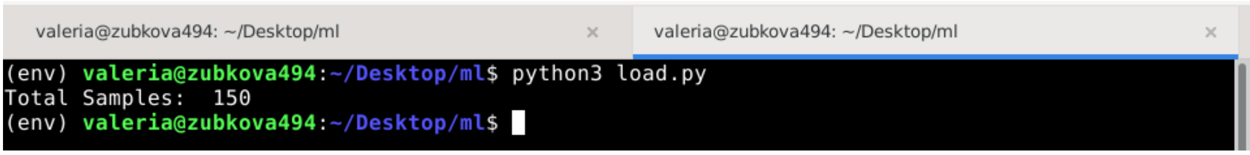


Рисунок 28 - Просмотр файла iris.onnx.


~/Desktop/ml/load.py - Mousepad (on zubkova494)
File Edit Search View Document Help
redis.conf
redis.log
iris.csv
model.py
load.py

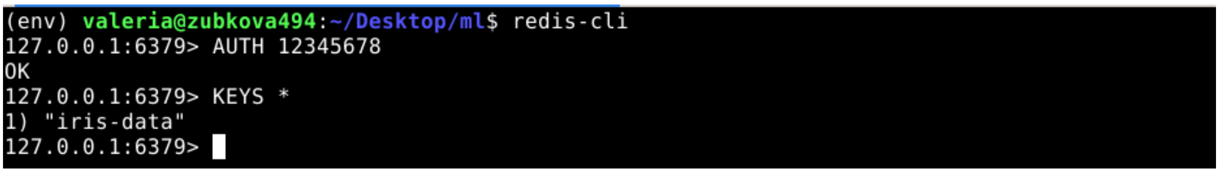
```python
import numpy as np

dataset = np.loadtxt('iris.csv', delimiter=',')
print("Total Samples: ", len(dataset))

import redis

red = redis.Redis(
        host='127.0.0.1',
        port='6379',
        password='12345678'
)

for i in range(0, len(dataset)):
        sl = dataset[i, 0]
        sq = dataset[i, 1]
        pl = dataset[i, 2]
        pw = dataset[i, 3]

        red.rpush('iris-data', f"{sl} {sq} {pl} {pw}")
```

Рисунок 29 - Содержимое скрипта load.py.


valeria@zubkova494: ~/Desktop/ml
valeria@zubkova494: ~/Desktop/ml

```
(env) valeria@zubkova494:~/Desktop/ml$ python3 load.py
Total Samples:  150
(env) valeria@zubkova494:~/Desktop/ml$ 
```

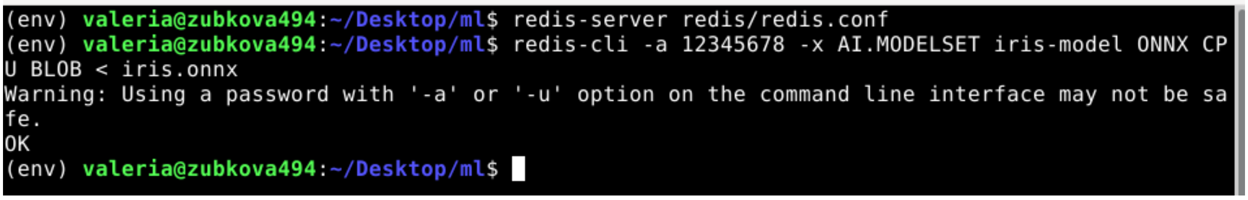
Рисунок 30 - Запуск файла load.py



```
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli
127.0.0.1:6379> AUTH 12345678
OK
127.0.0.1:6379> KEYS *
1) "iris-data"
127.0.0.1:6379> 
```

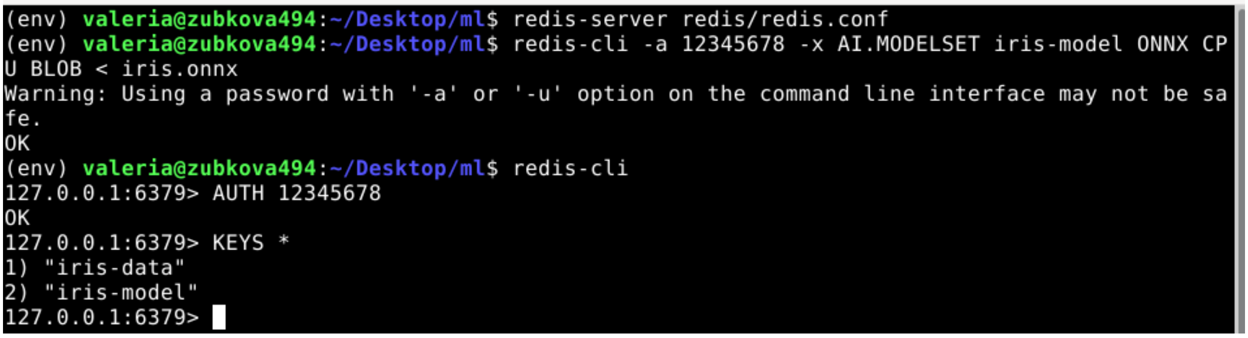
Рисунок 31 - Проверка какие есть ключи в Redis

32



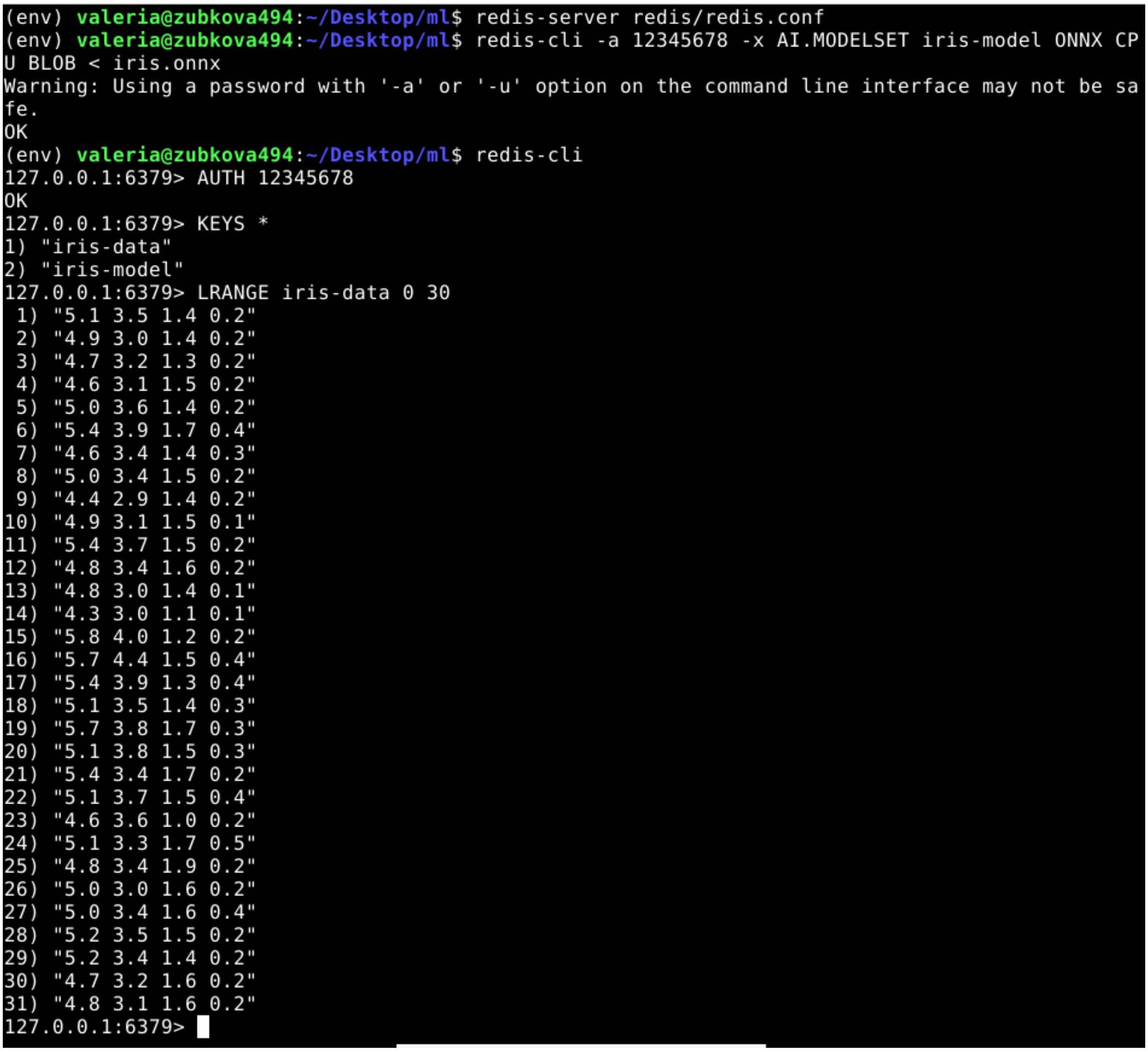
```
(env) valeria@zubkova494:~/Desktop/ml$ redis-server redis/redis.conf
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli -a 12345678 -x AI.MODELSET iris-model ONNX CP
U BLOB < iris.onnx
Warning: Using a password with '-a' or '-u' option on the command line interface may not be sa
fe.
OK
(env) valeria@zubkova494:~/Desktop/ml$ 
```

Рисунок 32 - Создание модели iris-model из файла iris.onnx.



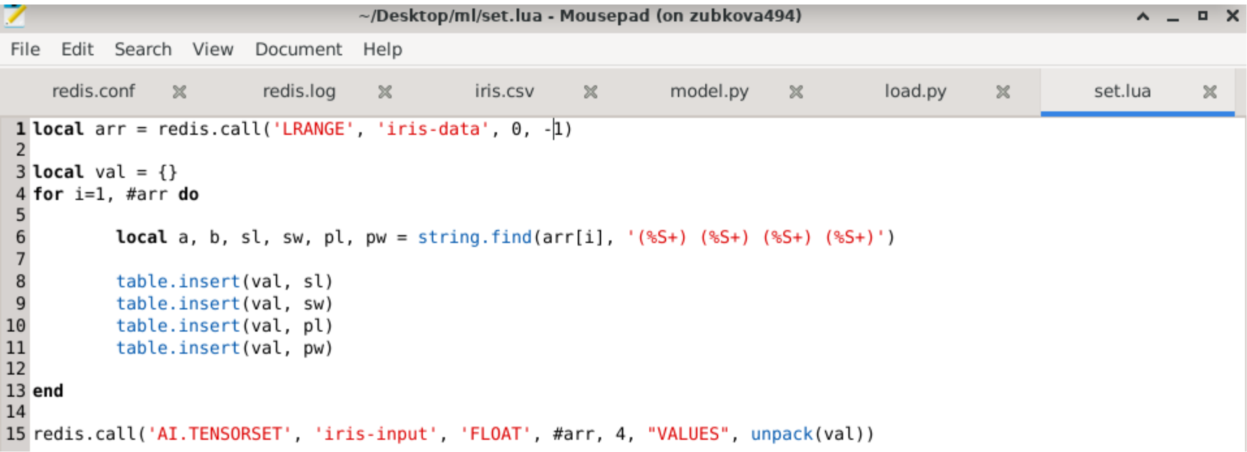
```
(env) valeria@zubkova494:~/Desktop/ml$ redis-server redis/redis.conf
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli -a 12345678 -x AI.MODELSET iris-model ONNX CP
U BLOB < iris.onnx
Warning: Using a password with '-a' or '-u' option on the command line interface may not be sa
fe.
OK
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli
127.0.0.1:6379> AUTH 12345678
OK
127.0.0.1:6379> KEYS *
1) "iris-data"
2) "iris-model"
127.0.0.1:6379> 
```

Рисунок 33 - Создание модели iris-model из файла iris.onnx.



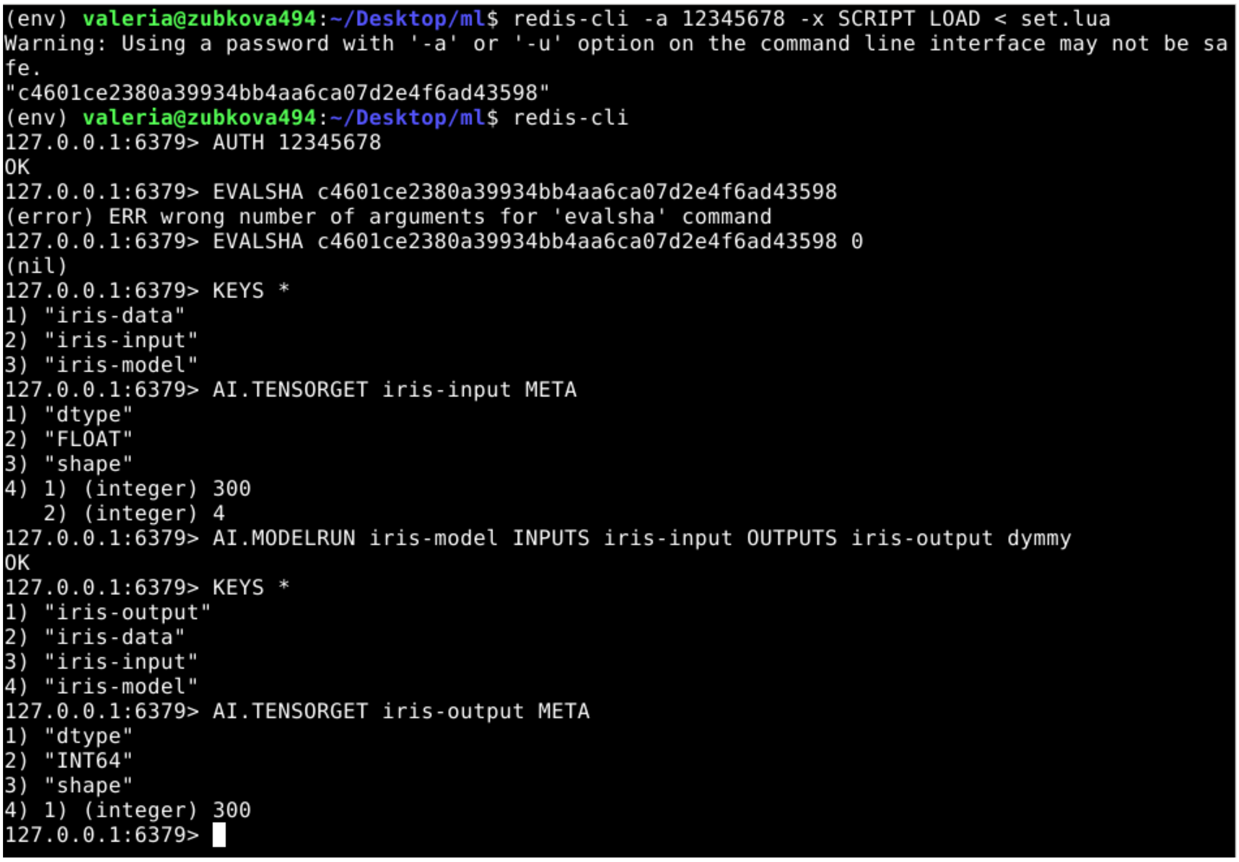
```
(env) valeria@zubkova494:~/Desktop/ml$ redis-server redis/redis.conf
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli -a 12345678 -x AI.MODELSET iris-model ONNX CP
U BLOB < iris.onnx
Warning: Using a password with '-a' or '-u' option on the command line interface may not be sa
fe.
OK
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli
127.0.0.1:6379> AUTH 12345678
OK
127.0.0.1:6379> KEYS *
1) "iris-data"
2) "iris-model"
127.0.0.1:6379> LRANGE iris-data 0 30
 1) "5.1 3.5 1.4 0.2"
 2) "4.9 3.0 1.4 0.2"
 3) "4.7 3.2 1.3 0.2"
 4) "4.6 3.1 1.5 0.2"
 5) "5.0 3.6 1.4 0.2"
 6) "5.4 3.9 1.7 0.4"
 7) "4.6 3.4 1.4 0.3"
 8) "5.0 3.4 1.5 0.2"
 9) "4.4 2.9 1.4 0.2"
10) "4.9 3.1 1.5 0.1"
11) "5.4 3.7 1.5 0.2"
12) "4.8 3.4 1.6 0.2"
13) "4.8 3.0 1.4 0.1"
14) "4.3 3.0 1.1 0.1"
15) "5.8 4.0 1.2 0.2"
16) "5.7 4.4 1.5 0.4"
17) "5.4 3.9 1.3 0.4"
18) "5.1 3.5 1.4 0.3"
19) "5.7 3.8 1.7 0.3"
20) "5.1 3.8 1.5 0.3"
21) "5.4 3.4 1.7 0.2"
22) "5.1 3.7 1.5 0.4"
23) "4.6 3.6 1.0 0.2"
24) "5.1 3.3 1.7 0.5"
25) "4.8 3.4 1.9 0.2"
26) "5.0 3.0 1.6 0.2"
27) "5.0 3.4 1.6 0.4"
28) "5.2 3.5 1.5 0.2"
29) "5.2 3.4 1.4 0.2"
30) "4.7 3.2 1.6 0.2"
31) "4.8 3.1 1.6 0.2"
127.0.0.1:6379> 
```

Рисунок 34 - Проверка создания объекта iris-data и вывод примера содержимого.



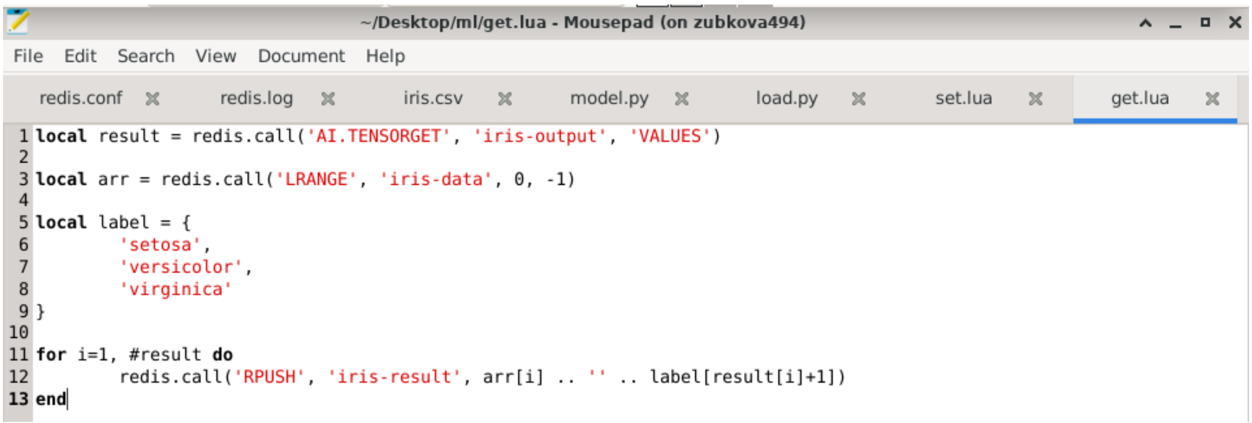
```lua
local arr = redis.call('LRANGE', 'iris-data', 0, -1)

local val = {}
for i=1, #arr do

        local a, b, sl, sw, pl, pw = string.find(arr[i], '(%S+) (%S+) (%S+) (%S+)')

        table.insert(val, sl)
        table.insert(val, sw)
        table.insert(val, pl)
        table.insert(val, pw)

end

redis.call('AI.TENSORSET', 'iris-input', 'FLOAT', #arr, 4, "VALUES", unpack(val))
```

Рисунок 35 - Содержимое скрипта set.lua.



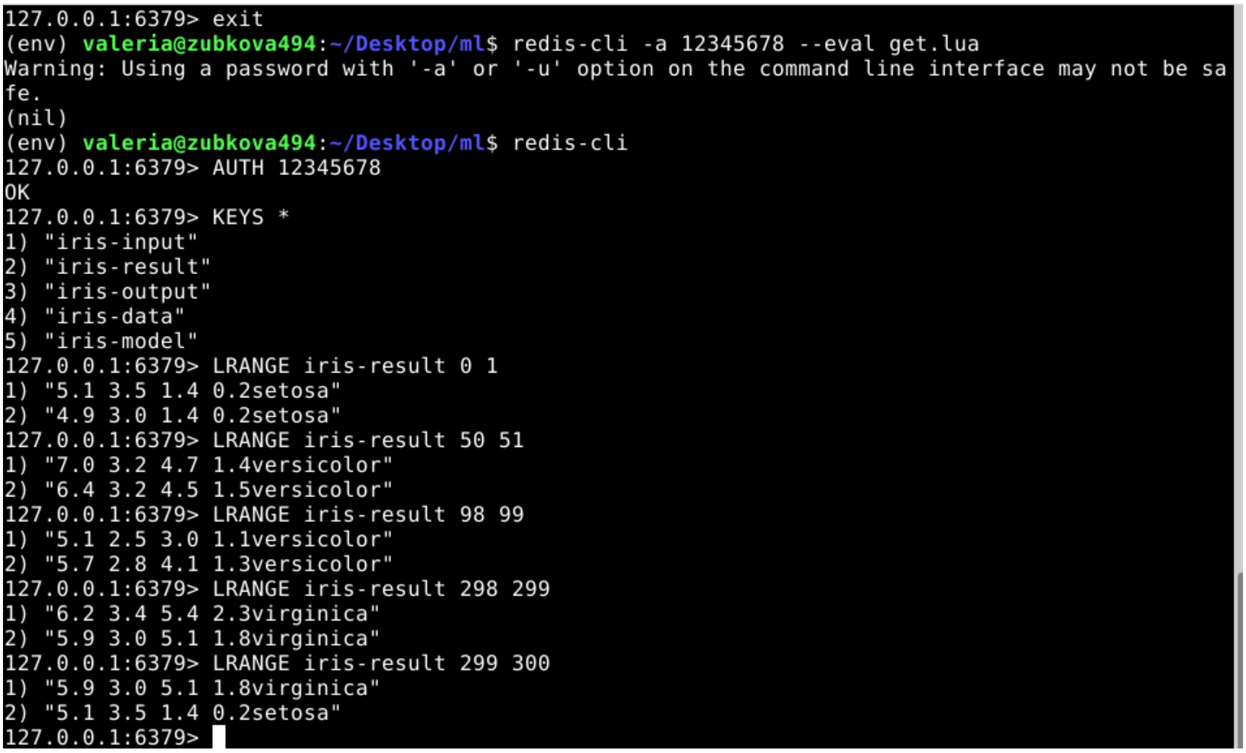
```
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli -a 12345678 -x SCRIPT LOAD < set.lua
Warning: Using a password with '-a' or '-u' option on the command line interface may not be sa
fe.
"c4601ce2380a39934bb4aa6ca07d2e4f6ad43598"
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli
127.0.0.1:6379> AUTH 12345678
OK
127.0.0.1:6379> EVALSHA c4601ce2380a39934bb4aa6ca07d2e4f6ad43598
(error) ERR wrong number of arguments for 'evalsha' command
127.0.0.1:6379> EVALSHA c4601ce2380a39934bb4aa6ca07d2e4f6ad43598 0
(nil)
127.0.0.1:6379> KEYS *
1) "iris-data"
2) "iris-input"
3) "iris-model"
127.0.0.1:6379> AI.TENSORGET iris-input META
1) "dtype"
2) "FLOAT"
3) "shape"
4) 1) (integer) 300
   2) (integer) 4
127.0.0.1:6379> AI.MODELRUN iris-model INPUTS iris-input OUTPUTS iris-output dymmy
OK
127.0.0.1:6379> KEYS *
1) "iris-output"
2) "iris-data"
3) "iris-input"
4) "iris-model"
127.0.0.1:6379> AI.TENSORGET iris-output META
1) "dtype"
2) "INT64"
3) "shape"
4) 1) (integer) 300
127.0.0.1:6379> 
```

Рисунок 36 - Выполнение скрипта set.lua и проверка создания объекта iris-input


~/Desktop/ml/get.lua - Mousepad (on zubkova494)
File Edit Search View Document Help
redis.conf
redis.log
iris.csv
model.py
load.py
set.lua
get.lua

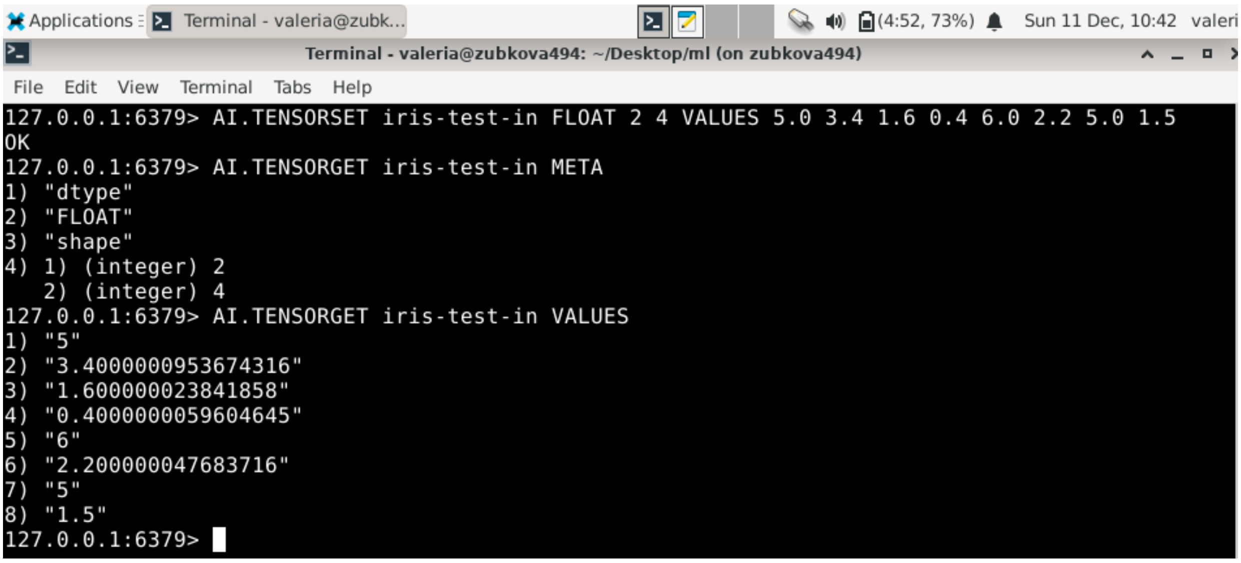
```
local result = redis.call('AI.TENSORGET', 'iris-output', 'VALUES')

local arr = redis.call('LRANGE', 'iris-data', 0, -1)

local label = {
        'setosa',
        'versicolor',
        'virginica'
}

for i=1, #result do
        redis.call('RPUSH', 'iris-result', arr[i] .. '' .. label[result[i]+1])
end
```

Рисунок 37 - Содержимое get.lua



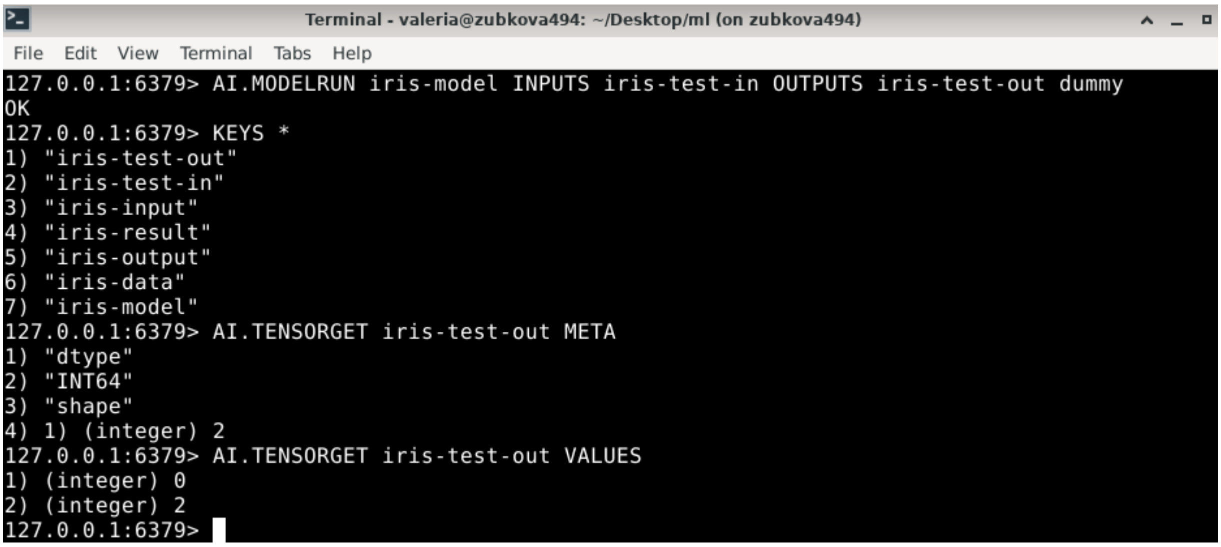
```
127.0.0.1:6379> exit
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli -a 12345678 --eval get.lua
Warning: Using a password with '-a' or '-u' option on the command line interface may not be sa
fe.
(nil)
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli
127.0.0.1:6379> AUTH 12345678
OK
127.0.0.1:6379> KEYS *
1) "iris-input"
2) "iris-result"
3) "iris-output"
4) "iris-data"
5) "iris-model"
127.0.0.1:6379> LRANGE iris-result 0 1
1) "5.1 3.5 1.4 0.2setosa"
2) "4.9 3.0 1.4 0.2setosa"
127.0.0.1:6379> LRANGE iris-result 50 51
1) "7.0 3.2 4.7 1.4versicolor"
2) "6.4 3.2 4.5 1.5versicolor"
127.0.0.1:6379> LRANGE iris-result 98 99
1) "5.1 2.5 3.0 1.1versicolor"
2) "5.7 2.8 4.1 1.3versicolor"
127.0.0.1:6379> LRANGE iris-result 298 299
1) "6.2 3.4 5.4 2.3virginica"
2) "5.9 3.0 5.1 1.8virginica"
127.0.0.1:6379> LRANGE iris-result 299 300
1) "5.9 3.0 5.1 1.8virginica"
2) "5.1 3.5 1.4 0.2setosa"
127.0.0.1:6379>
```

Рисунок 38 - Выполнение скрипта get.lua


Applications
Terminal - valeria@zubk...
(4:52, 73%)
Sun 11 Dec, 10:42
valeri
Terminal - valeria@zubkova494: ~/Desktop/ml (on zubkova494)
File Edit View Terminal Tabs Help

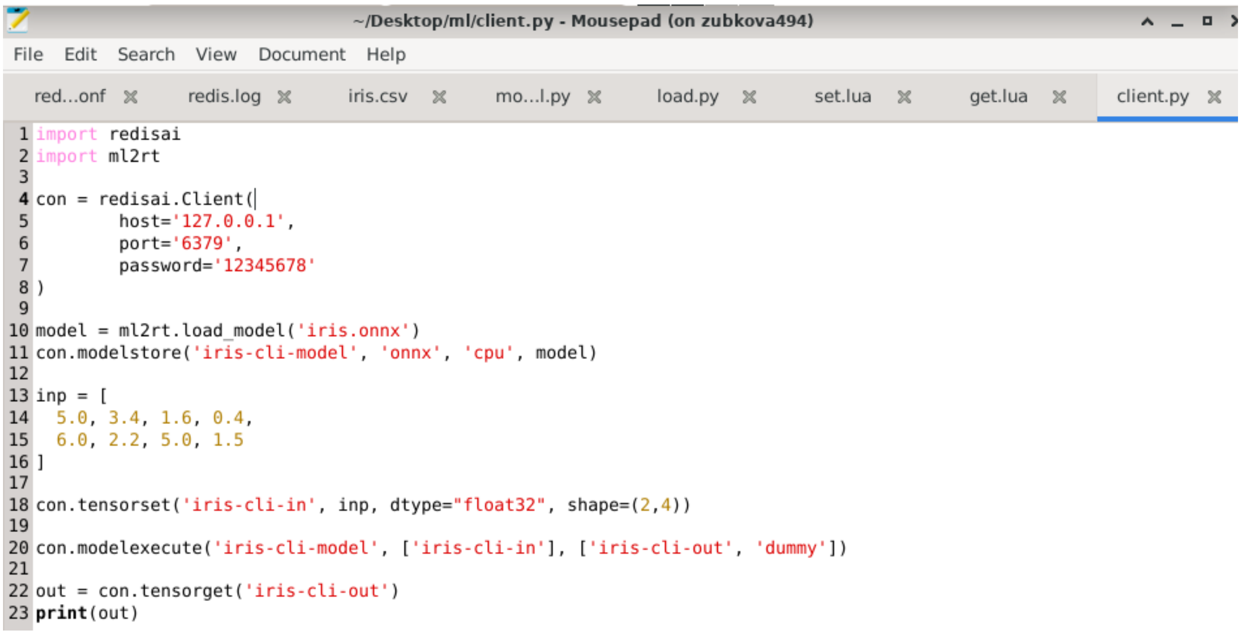
```
127.0.0.1:6379> AI.TENSORSET iris-test-in FLOAT 2 4 VALUES 5.0 3.4 1.6 0.4 6.0 2.2 5.0 1.5
OK
127.0.0.1:6379> AI.TENSORGET iris-test-in META
1) "dtype"
2) "FLOAT"
3) "shape"
4) 1) (integer) 2
   2) (integer) 4
127.0.0.1:6379> AI.TENSORGET iris-test-in VALUES
1) "5"
2) "3.4000000953674316"
3) "1.600000023841858"
4) "0.4000000059604645"
5) "6"
6) "2.200000047683716"
7) "5"
8) "1.5"
127.0.0.1:6379>
```

Рисунок 39 - Создание тестового набора данных.


Terminal - valeria@zubkova494: ~/Desktop/ml (on zubkova494)
File Edit View Terminal Tabs Help

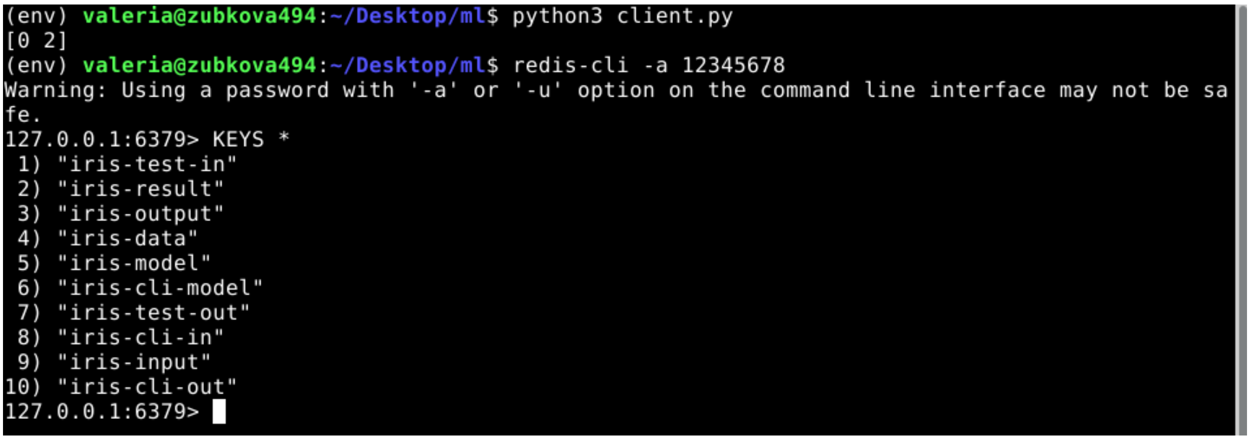
```
127.0.0.1:6379> AI.MODELRUN iris-model INPUTS iris-test-in OUTPUTS iris-test-out dummy
OK
127.0.0.1:6379> KEYS *
1) "iris-test-out"
2) "iris-test-in"
3) "iris-input"
4) "iris-result"
5) "iris-output"
6) "iris-data"
7) "iris-model"
127.0.0.1:6379> AI.TENSORGET iris-test-out META
1) "dtype"
2) "INT64"
3) "shape"
4) 1) (integer) 2
127.0.0.1:6379> AI.TENSORGET iris-test-out VALUES
1) (integer) 0
2) (integer) 2
127.0.0.1:6379>
```

Рисунок 40 - Создание объекта iris-test-out на основе тестовых данных.


~/Desktop/ml/client.py - Mousepad (on zubkova494)
File Edit Search View Document Help
red...onf
redis.log
iris.csv
mo...l.py
load.py
set.lua
get.lua
client.py

```python
import redisai
import ml2rt

con = redisai.Client(
        host='127.0.0.1',
        port='6379',
        password='12345678'
)

model = ml2rt.load_model('iris.onnx')
con.modelstore('iris-cli-model', 'onnx', 'cpu', model)

inp = [
  5.0, 3.4, 1.6, 0.4,
  6.0, 2.2, 5.0, 1.5
]

con.tensorset('iris-cli-in', inp, dtype="float32", shape=(2,4))

con.modelexecute('iris-cli-model', ['iris-cli-in'], ['iris-cli-out', 'dummy'])

out = con.tensorget('iris-cli-out')
print(out)
```

Рисунок 41 - Содержимое скрипта client.py.



```
(env) valeria@zubkova494:~/Desktop/ml$ python3 client.py
[0 2]
(env) valeria@zubkova494:~/Desktop/ml$ redis-cli -a 12345678
Warning: Using a password with '-a' or '-u' option on the command line interface may not be sa
fe.
127.0.0.1:6379> KEYS *
 1) "iris-test-in"
 2) "iris-result"
 3) "iris-output"
 4) "iris-data"
 5) "iris-model"
 6) "iris-cli-model"
 7) "iris-test-out"
 8) "iris-cli-in"
 9) "iris-input"
10) "iris-cli-out"
127.0.0.1:6379> 
```

Рисунок 42 - Выполнение скрипта client.py и проверка его выполнения.