**Experiment-1:-**

**Aim of the experiment:- Write a program find the crop yield of different region.**

**Code:-**

**{**

**"cells": [**

**{**

**"cell\_type": "code",**

**"execution\_count": 15,**

**"id": "324855ac",**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"w1, w2, w3 = 0.3, 0.2, 0.5"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 16,**

**"id": "62ec9f2a",**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"kanto\_tem = 73\n",**

**"kanto\_rainfall = 67\n",**

**"kanto\_humidity = 43"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 17,**

**"id": "1b2eceab",**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/plain": [**

**"56.8"**

**]**

**},**

**"execution\_count": 17,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"kanto\_yield\_apples = kanto\_tem \* w1 + kanto\_rainfall \* w2 + kanto\_humidity \* w3\n",**

**"kanto\_yield\_apples"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 18,**

**"id": "7f243206",**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"kanto = [73,67,43]\n",**

**"johto = [91,88,64]\n",**

**"hoenn = [87,134,58]\n",**

**"sinonh = [102,43,37]\n",**

**"unova = [69,96,70]"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 19,**

**"id": "0ce22cbe",**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"weights = [w1, w2, w3]"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 24,**

**"id": "b616c6e3",**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"def crop\_yield(region,weights):\n",**

**" result = 0\n",**

**" for x, w in zip(region, weights):\n",**

**" result += x\*w #RESULT =RESULT+X\*W\n",**

**" return result"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 25,**

**"id": "283bd179",**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/plain": [**

**"76.9"**

**]**

**},**

**"execution\_count": 25,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"crop\_yield(johto, weights)\n"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 26,**

**"id": "0470b2cd",**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/plain": [**

**"56.8"**

**]**

**},**

**"execution\_count": 26,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"crop\_yield(kanto, weights)"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 27,**

**"id": "bb908280",**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/plain": [**

**"81.9"**

**]**

**},**

**"execution\_count": 27,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

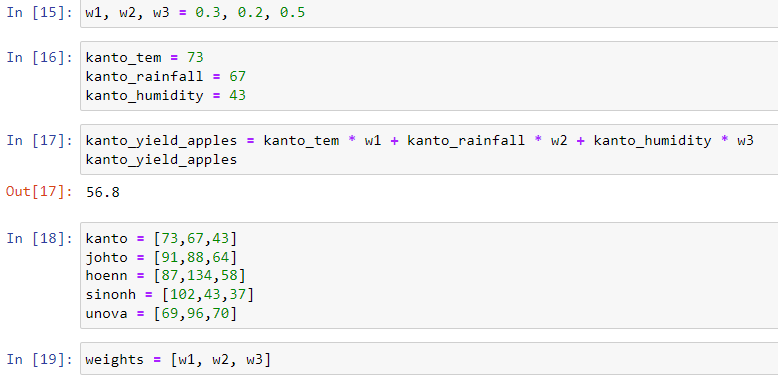
**"source": [**

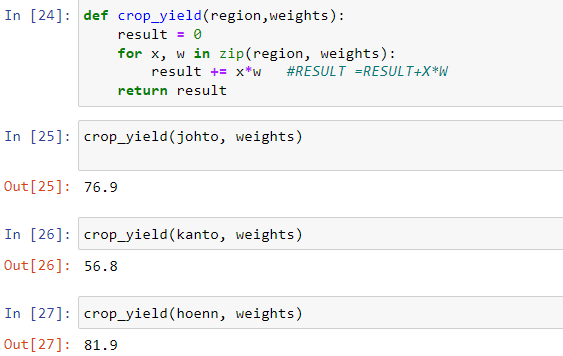
**"crop\_yield(hoenn, weights)"**

**]**

**},**

**Output:-**

****



**Conclusion:-**successfully find the crop yield in different region.

**Experiment:2**

**Aim of the experiment:-How to install package.**

**Code:-**

**{**

**"cell\_type": "code",**

**"execution\_count": 28,**

**"id": "79dabc2e",**

**"metadata": {},**

**"outputs": [**

**{**

**"name": "stdout",**

**"output\_type": "stream",**

**"text": [**

**"Requirement already satisfied: pip in c:\\users\\gyan\\anaconda3\\lib\\site-packages (21.3.1)\n"**

**]**

**},**

**{**

**"name": "stderr",**

**"output\_type": "stream",**

**"text": [**

**"WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip.\_vendor.urllib3.connection.HTTPSConnection object at 0x0000026E2EB93AF0>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/pip/\n",**

**"WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip.\_vendor.urllib3.connection.HTTPSConnection object at 0x0000026E2EB93D00>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/pip/\n",**

**"WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip.\_vendor.urllib3.connection.HTTPSConnection object at 0x0000026E2EB93EB0>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/pip/\n",**

**"WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip.\_vendor.urllib3.connection.HTTPSConnection object at 0x0000026E2EBC30A0>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/pip/\n",**

**"WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip.\_vendor.urllib3.connection.HTTPSConnection object at 0x0000026E2EBC3250>: Failed to establish a new connection: [Errno 11001] getaddrinfo failed')': /simple/pip/\n"**

**]**

**}**

**],**

**"source": [**

**"#install pip\n",**

**"!python -m pip install --upgrade pip"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 29,**

**"id": "680a0561",**

**"metadata": {},**

**"outputs": [**

**{**

**"name": "stdout",**

**"output\_type": "stream",**

**"text": [**

**"\n",**

**"Usage: \n",**

**" pip <command> [options]\n",**

**"\n",**

**"Commands:\n",**

**" install Install packages.\n",**

**" download Download packages.\n",**

**" uninstall Uninstall packages.\n",**

**" freeze Output installed packages in requirements format.\n",**

**" list List installed packages.\n",**

**" show Show information about installed packages.\n",**

**" check Verify installed packages have compatible dependencies.\n",**

**" config Manage local and global configuration.\n",**

**" search Search PyPI for packages.\n",**

**" cache Inspect and manage pip's wheel cache.\n",**

**" index Inspect information available from package indexes.\n",**

**" wheel Build wheels from your requirements.\n",**

**" hash Compute hashes of package archives.\n",**

**" completion A helper command used for command completion.\n",**

**" debug Show information useful for debugging.\n",**

**" help Show help for commands.\n",**

**"\n",**

**"General Options:\n",**

**" -h, --help Show help.\n",**

**" --debug Let unhandled exceptions propagate outside the\n",**

**" main subroutine, instead of logging them to\n",**

**" stderr.\n",**

**" --isolated Run pip in an isolated mode, ignoring\n",**

**" environment variables and user configuration.\n",**

**" -v, --verbose Give more output. Option is additive, and can be\n",**

**" used up to 3 times.\n",**

**" -V, --version Show version and exit.\n",**

**" -q, --quiet Give less output. Option is additive, and can be\n",**

**" used up to 3 times (corresponding to WARNING,\n",**

**" ERROR, and CRITICAL logging levels).\n",**

**" --log <path> Path to a verbose appending log.\n",**

**" --no-input Disable prompting for input.\n",**

**" --proxy <proxy> Specify a proxy in the form\n",**

**" [user:passwd@]proxy.server:port.\n",**

**" --retries <retries> Maximum number of retries each connection should\n",**

**" attempt (default 5 times).\n",**

**" --timeout <sec> Set the socket timeout (default 15 seconds).\n",**

**" --exists-action <action> Default action when a path already exists:\n",**

**" (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.\n",**

**" --trusted-host <hostname> Mark this host or host:port pair as trusted,\n",**

**" even though it does not have valid or any HTTPS.\n",**

**" --cert <path> Path to PEM-encoded CA certificate bundle. If\n",**

**" provided, overrides the default. See 'SSL\n",**

**" Certificate Verification' in pip documentation\n",**

**" for more information.\n",**

**" --client-cert <path> Path to SSL client certificate, a single file\n",**

**" containing the private key and the certificate\n",**

**" in PEM format.\n",**

**" --cache-dir <dir> Store the cache data in <dir>.\n",**

**" --no-cache-dir Disable the cache.\n",**

**" --disable-pip-version-check\n",**

**" Don't periodically check PyPI to determine\n",**

**" whether a new version of pip is available for\n",**

**" download. Implied with --no-index.\n",**

**" --no-color Suppress colored output.\n",**

**" --no-python-version-warning\n",**

**" Silence deprecation warnings for upcoming\n",**

**" unsupported Pythons.\n",**

**" --use-feature <feature> Enable new functionality, that may be backward\n",**

**" incompatible.\n",**

**" --use-deprecated <feature> Enable deprecated functionality, that will be\n",**

**" removed in the future.\n"**

**]**

**}**

**],**

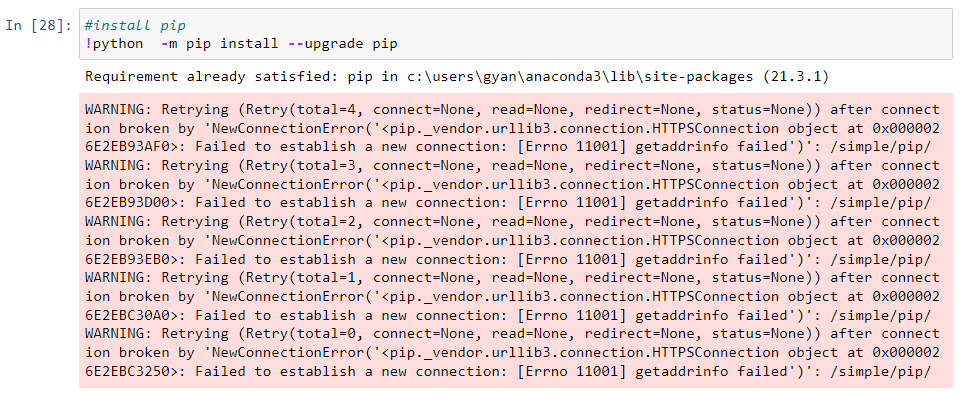
**"source": [**

**"!pip -v"**

**]**

**},**

**{**

****

**Conclusion:- successfully install package by pip command.**

**Experiment:-3**

**Aim:- Install numpy using pip command**

**Code:-**

**"cell\_type": "code",**

**"execution\_count": 30,**

**"id": "772e73ae",**

**"metadata": {},**

**"outputs": [**

**{**

**"name": "stderr",**

**"output\_type": "stream",**

**"text": [**

**"\n",**

**"Usage: \n",**

**" pip install [options] <requirement specifier> [package-index-options] ...\n",**

**" pip install [options] -r <requirements file> [package-index-options] ...\n",**

**" pip install [options] [-e] <vcs project url> ...\n",**

**" pip install [options] [-e] <local project path> ...\n",**

**" pip install [options] <archive url/path> ...\n",**

**"\n",**

**"no such option: --quite\n"**

**]**

**}**

**],**

**"source": [**

**"#pip install python packages in any environment\n",**

**"!pip install numpy --upgrade --quite --user"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 31,**

**"id": "ca011df5",**

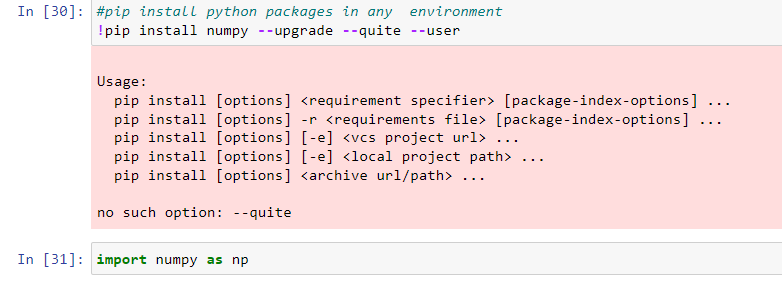
**"metadata": {},**

**"outputs": [],**

**"source": [**

**"import numpy as np"**

**Output:-**



**Conclusion:-**By doing above experiment we successfully install numpy and start using.

**Experiment:-4**

**Aim:- find the crop yield and weight of any region by using numpy.**

**Code:-**

**{**

**"cell\_type": "code",**

**"execution\_count": 32,**

**"id": "8c3a3b39",**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"kanto =np.array([73,67,43])"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 33,**

**"id": "b6102d86",**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/plain": [**

**"array([73, 67, 43])"**

**]**

**},**

**"execution\_count": 33,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"kanto"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 34,**

**"id": "d160b6ad",**

**"metadata": {},**

**"outputs": [],**

**"source": [**

**"weights =np.array([w1,w2,w3])"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 35,**

**"id": "9c7436f3",**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/plain": [**

**"array([0.3, 0.2, 0.5])"**

**]**

**},**

**"execution\_count": 35,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

**"source": [**

**"weights"**

**]**

**},**

**{**

**"cell\_type": "code",**

**"execution\_count": 36,**

**"id": "72737f6d",**

**"metadata": {},**

**"outputs": [**

**{**

**"data": {**

**"text/plain": [**

**"0.3"**

**]**

**},**

**"execution\_count": 36,**

**"metadata": {},**

**"output\_type": "execute\_result"**

**}**

**],**

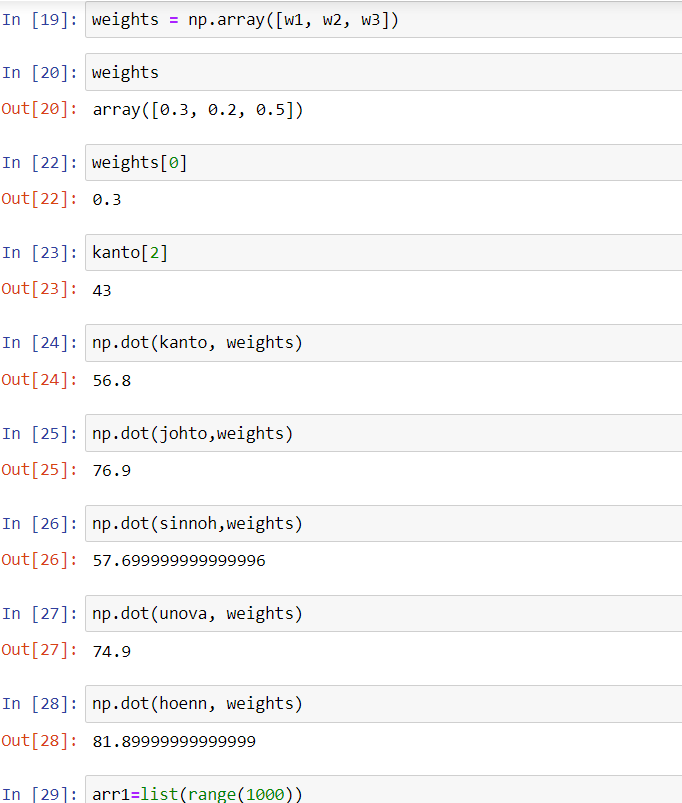
**"source": [**

**"weights[0]"**

**]**

**},**

**{**



**Conclusion:-** By doing above experiment we successfully find weight of individually .