

# INTERNSHIP REPORT: DLITHE –Machine Learning with Python

---

Submitted by:- Aravinda Krishna U

3<sup>rd</sup> year B.E. Computer Science & Engineering

NMAM Institute of Technology



# Table of Contents



**1.Acknowledg  
ement**



**2.About the  
company**



**3.Summary**



**4.Internship  
Contents**

ML

Python



**5. Conclusion**

# Acknowledgement

I would like to thank MS. Prachi Tare for her guidance during the period of the internship. She has been very approachable and has always been available to clarify any doubts that I had regarding my internship. I would also like to express my gratitude to Mr. Arun Rajpurohit for helping us coordinates both during the live sessions and out of it and enhancing the experience.

# About the company

- DLithe is a consultancy service company from Bangalore.
- They work towards developing an agile workforce through placement training, hackathons and internship platforms.
- They assist clients to upskill in a variety of computer related activities such as product development, coding, cyber security, etc.
- They are the people from Industry. They have experience working with customers globally to achieve their business goals.
- They have worked with many large enterprise engagements in various domains like Retail, Manufacturing, Banking & Finance, Insurance, etc, using various technologies like Java, Microsoft, Python, AI, IoT and many more...Legacy to Modern Technology.
- They have also taken up the opportunity to guide students to learn the know-how of the working of a software company. They encourage students to learn and apply what they learn.

# Summary

I did an internship in DLithe on Machine Learning with Python from 25 October 2021 to 25 November 2021. During this period, I gained knowledge on the various aspects of Python and its libraries. I have learnt many basics about Python. I learnt how the basic python is necessary to learn Machine learning and got good hold about python with clear conceptual understanding. We started with basics of python like python syntax, variables, datatypes, operators, lists, dictionaries, tuples. We learnt about different control statements like if, if else, nested if. We also learnt about for loop, while loop statements and also different functions on list, array, dictionary. We learnt how to use pandas, NumPy, matplotlib. Using these concepts, we implemented Machine learning.

# Machine Learning:


- Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.
- Machine learning is an important component of the growing field of data science.
- Using statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects.
- These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics.
- As big data continues to expand and grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them.

# Python:

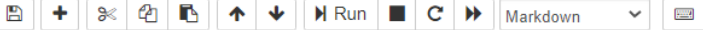
- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.
- Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.
- Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.
- Python supports modules and packages, which encourages program modularity and code reuse.
- The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.
- We have used Jupyter Notebook in this course to run Python codes

# Steps involved in python course:

## 1. Data types

jupyter Basic\_to\_data\_type (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 C



### Data Types

```
In [ ]: #1. Integer
type(1)
Out[3]: int

In [ ]: type(1.23)
Out[4]: float

In [ ]: type(3j + 1)
Out[5]: complex

In [ ]: type('prachi')
Out[6]: str

In [ ]: type('1')
Out[8]: str

In [ ]: [1, 2, 3, 4, 5.60, 'ab']
Out[9]: [1, 2, 3, 4, 5.6, 'ab']

In [ ]: type([1, 2, 3, 4, 5.60, 'ab'])
Out[10]: list


In [ ]: #Dictionary
characters - keys
words - values

In [ ]: {'Employee_Name': 'Raju', 'Age': 26}
Out[11]: {'Employee_Name': 'Raju', 'Age': 26}

In [ ]: type({'Employee_Name': 'Raju', 'Age': 26})
Out[12]: dict
```



# 2. Variables

jupyter Basic\_to\_data\_type (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Save + Undo Copy Paste Up Down Run Stop Restart Markdown

```
type()

-----
TypeError                                Traceback (most recent call last)
<ipython-input-20-59467681e320> in <module>
      1 false = 2
----> 2 type()

TypeError: type() takes 1 or 3 arguments
```

In [ ]: `print(1)`

1

In [ ]: `print("What a day! It was good but it could be better tomorrow")`

What a day! It was good but it could be better tomorrow

In [ ]: `L = [1,2,3,4,5,6,7,8,9,10]`  
`print('The list is', L)`

The list is [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]


## Variables

## Decorators











In [ ]: `1. You can right Upper and lower case letter first`  
`2. You can also mention the integers`  
`3. You can also use underscore`

`1. You cannot use special characters`  
`2. You cannot use Space bar`  
`3. You cannot declare the variable starting with an interger`

# 3. Lists

jupyter Python\_Operators (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

      Run    Code 

## Python List

```
In [ ]: 1. append

In [14]: ap = [1,2,3,4,5]
         ap.append()

In [15]: print(ap)
         [1, 2, 3, 4, 5, 10]

In [ ]: 2. Insert

In [16]: ins = [4,8,20,12,2,6]
         ins.insert(3,5.4)

In [17]: print(ins)
         [4, 8, 20, 5.4, 12, 2, 6]

In [19]: i = [2,3,5,34,53,72,624,7,47,28,3,724,72,726,247,24,72,472,72,4724,72,47,24,8]

In [20]: i.insert(11,'ab')

In [21]: print(i)
         [2, 3, 5, 34, 53, 72, 624, 7, 47, 28, 3, 'ab', 724, 72, 726, 247, 24, 72, 472, 72, 4724, 72, 47, 24, 8]

In [22]: i.insert(5,'class')

In [24]: print(i)
         [2, 3, 5, 34, 53, 'class', 72, 624, 7, 47, 28, 3, 'ab', 724, 72, 726, 247, 24, 72, 472, 72, 4724, 72, 47, 24, 8]

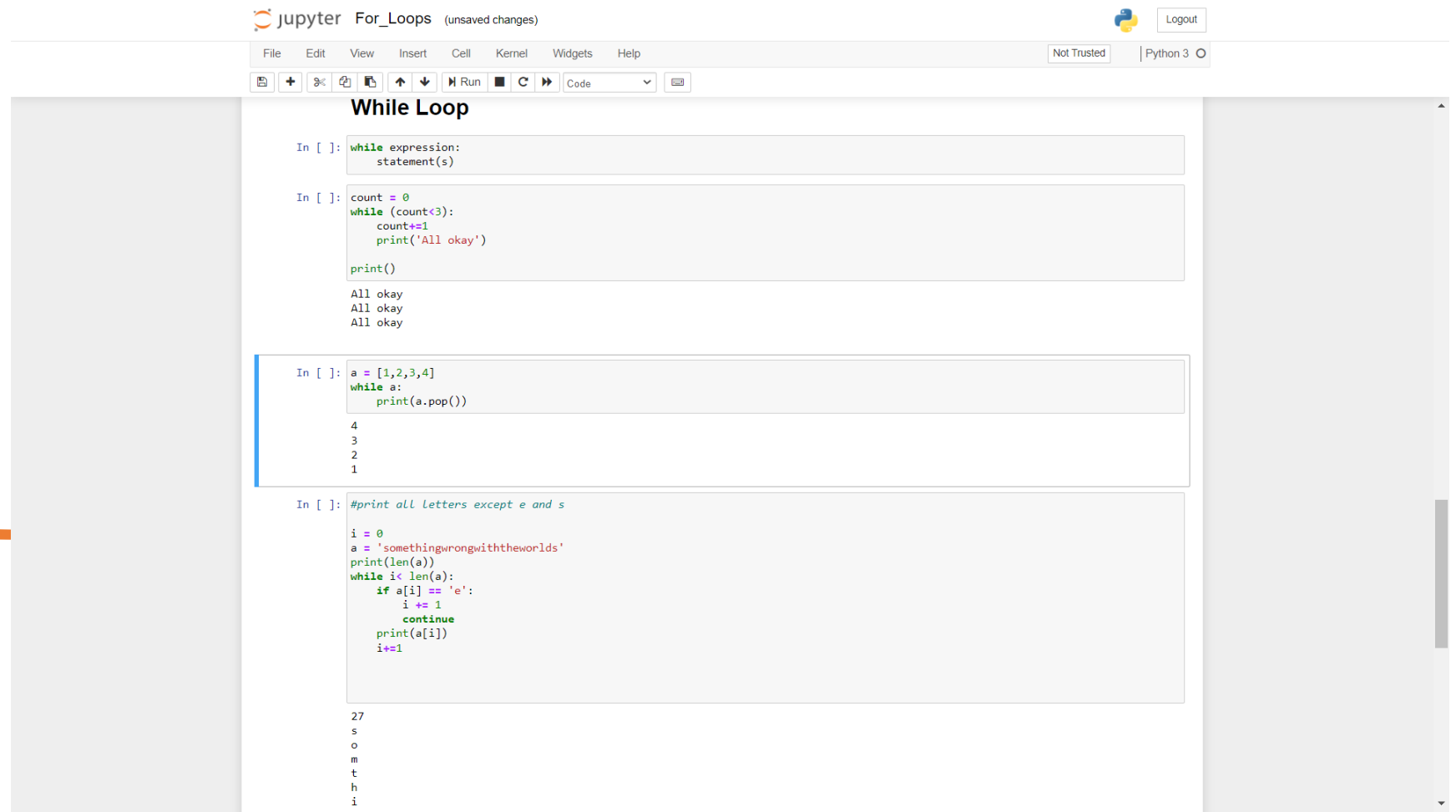
In [ ]: 3. Extend

In [25]: ex1 = [2,3,4,7]
         ex2 = [78,4,8,12]

In [26]: ex1.extend(ex2)

In [27]: print(ex1)
```

# 4. Loops



The image shows a Jupyter Notebook interface with the title "For\_Loops (unsaved changes)". The notebook is running on Python 3. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and saving. The notebook content is titled "While Loop" and contains three code cells.

```
In [ ]: while expression:
        statement(s)
```

```
In [ ]: count = 0
while (count<3):
    count+=1
    print('All okay')

print()

All okay
All okay
All okay
```

```
In [ ]: a = [1,2,3,4]
while a:
    print(a.pop())

4
3
2
1
```

```
In [ ]: #print all Letters except e and s

i = 0
a = 'somethingwrongwiththeworlds'
print(len(a))
while i< len(a):
    if a[i] == 'e':
        i += 1
        continue
    print(a[i])
    i+=1

27
s
o
m
t
h
i
```

# 5.Statement

jupyter Functions (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3

Run Code

### Statement

```
In [ ]: # Declare using Continuation Character

s = 1 + 2 + 3 + \
    7 + 8 + 9 + \
    45 + 3

In [ ]: print(s)

78

In [ ]: s = "oeifhoifhwoiejgpiwejgpigoiwrgwpigjpwijgldpiwjrouehfpiwrhj[owjfoqwfwpiegj[oehkpotfjioehqiehjowhjpwojwpi]]"

In [ ]: # Using Paranthesis

h = ( 1 + 2 + 5 *4**6)
print(h)

20483

In [ ]: # Decvlare using braces

d = {54*45 +3}
print(d)

{2433}

In [ ]: {'s': [1,2,3,5]}

In [ ]: s = ['USA','China']

In [ ]: #Declared using semicolon

flog = 0;rope = 9; paper = 10

In [ ]: Executing statement:
        Statement1
        .....
        statement n
```

# 6.Lambda Functions

Jupyter Functions (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 C

Run Code

## Lambda Function

In [ ]: `lambda` function takes argument `as` such `as` we suggest but can only have one expression

`lambda` arguments : Expression

In [ ]: `# Adding the number five with any anumber mentioned or passed in the argument`

```
x = lambda a:a + 5
print(x(23))
```

28

In [ ]: `print(a)`

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-31-bca0e2660b9f> in <module>
----> 1 print(a)

NameError: name 'a' is not defined
```

In [ ]: `y = lambda a,b,c : a+b+c`

```
y(2,3,4)
```

Out[33]: 9

In [ ]: `b`

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-35-89e6c98d9288> in <module>
----> 1 b

NameError: name 'b' is not defined
```

In [ ]: `def myfunc(n):`  
`square = n**2`  
`return square`

```
print(myfunc(5))
```

25

# 7. String

## Strings

```
In [ ]: name = 'fIndtheCar'
```

```
In [ ]: name.capitalize()
```

```
Out[2]: 'Findthecar'
```

```
In [ ]: name.count('f')
```

```
Out[4]: 1
```

```
In [ ]: name.find('q')
```

```
Out[10]: -1
```

```
In [ ]: name.isalpha()
```

```
Out[11]: True
```

```
In [ ]: s = 'sdfbiqb8349dhjqr83yqdh033r'  
s.isalpha()
```

```
Out[13]: False
```

```
In [ ]: name.islower()
```

```
Out[14]: False
```

```
In [ ]: name.isupper()
```

```
Out[15]: False
```

```
In [ ]: name.lower()
```


```
Out[16]: 'findthecar'
```

```
In [ ]: a = "Python"  
a.split('y')
```

```
Out[21]: ['P', 'thon']
```

```
In [ ]: a*2
```

# 8. Tuple

jupyter Functions\_in\_data\_type (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Out[45]: [1, 2, 3, 4]

## Tuple

```
In [ ]: this1 = ('Banana','orange','kiwi','melon','mango','apple')
```

```
In [ ]: #INDEXING
this1[1]
```

```
Out[2]: 'orange'
```

```
In [ ]: #Slicing
this1[1:4]
```

```
Out[3]: ('orange', 'kiwi', 'melon')
```

```
In [ ]: tuple1 = (1,2,3,4)
tuple2 = ('AB',5,3,2)
tuple3 = tuple1 + tuple2
print(tuple3)

(1, 2, 3, 4, 'AB', 5, 3, 2)
```

```
In [ ]: # Creating a tuple
# With the help of builtin function
Tuple = tuple('Hello')
#What if I want input = ("Hello")
```

```
In [ ]: print(Tuple)

#What if I want input = ("Hello")

('H', 'e', 'l', 'l', 'o')
```


```
In [ ]: #Tuple Unpacking

tup = ('Hello','World',"Good")

a,b,c = tup
print(a)
print(b)
print(c)

Hello
World
Good
```

# 9.Dictionary

jupyter Functions\_in\_data\_type (unsaved changes)  Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 C

Good

## Dictionary

```
In [ ]: {key:pair,key1:pair1}
```

```
In [ ]: Dict = {}
Dict[0] = 'Apple'
Dict[1] = 'Kiwi'
Dict[2] = 2
```

```
In [ ]: print(Dict)
{0: 'Apple', 1: 'Kiwi', 2: 2}
```

```
In [ ]: dic = {'Name' : "Ajay", 'Age' : 26, 'Number' : 1284148}
```

```
In [ ]: dic['Name']
```

```
Out[60]: 'Ajay'
```

```
In [ ]: del dic['Name']
```

```
In [ ]: dic
```

```
Out[62]: {'Age': 26, 'Number': 1284148}
```

```
In [ ]: dic['Name'] = 'Ajay'
```

```
In [ ]: dic
```

```
Out[66]: {'Age': 26, 'Number': 1284148, 'Colon': 'red', 'Name': 'Ajay'}
```

```
In [ ]: b1 = dic.pop('Colon')
```

```
In [ ]: print(dic)
print(b1)
{'Age': 26, 'Number': 1284148, 'Name': 'Ajay'}
red
```

```
In [ ]: # constructor
abc = dict(brand = 'Ford', model = 'Mustang', year = 1964)
```



# 10. If, else-if and nested if statements

---

jupyter Input\_in\_Python (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

## If-Else condition

In [ ]: If Statement

It **is** the most simple kind of decision making forum where we decide whether to keep the statement **or not**

In [ ]: `if condition:  
 Statement 1  
statement 2`

In [21]: `i = 35  
if (i>25):  
 print(" 35 is less than 25")  
print("25 is less than 35")`

35 is less than 25  
25 is less than 35

In [ ]: `if (condition):  
 # execute the block of if  
 #condition is true  
  
else:  
 #execute the block of else  
 # condition is false`

In [23]: `i = 35  
if (i<25):  
 print(" 35 is less than 25")  
  
else:  
 print("25 is less than 35")`

25 is less than 35

In [ ]: `i = 10  
i <=15  
  
if and else condition. If condition is true print all okay and if it is false print try again`

In [27]: `x = int(input())  
r = x%2  
if r==0:  
 print("Even")  
|  
else:`

# 11.Numpy

jupyter Numpy\_Python (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 C

```
print(D.size*D.itemsize)
8000
```

## Operations in Numpy!

```
In [8]: a = np.array([(1,2,3),(4,5,6)])

In [9]: print(a)
[[1 2 3]
 [4 5 6]]

In [12]: a.sum(axis=1)
Out[12]: array([ 6, 15])

In [13]: np.sqrt(a)
Out[13]: array([[1.         , 1.41421356, 1.73205081],
                [2.         , 2.23606798, 2.44948974]])

In [14]: np.std(a)
Out[14]: 1.707825127659933

In [15]: b = np.array([(1,2,3),(3,4,5)])

In [16]: a+b
Out[16]: array([[ 2,  4,  6],
                [ 7,  9, 11]])

In [17]: a*b
Out[17]: array([[ 1,  4,  9],
                [12, 20, 30]])

In [18]: ar = np.array([1,2,3])

In [19]: ar
Out[19]: array([1, 2, 3])

In [20]: np.exp(ar)
```

# 12. Pandas

jupyter Pandas\_Python (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
In [ ]: import pandas as pd
import numpy as np
```

## Pandas

```
In [ ]: s = pd.Series([1,2,3,4,5])
```

```
In [ ]: print(s)
```

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

```
In [ ]: s1 = pd.Series(['Ajay', 'Suresh', 'Ram', 'Mohan'])
s2 = pd.Series([23, 25, 28, 32])
s3 = pd.Series([1, 2, 3, 4])
```

```
In [ ]: data = {'Name':s1, 'Age':s2, 'Roll_No':s3}
```

```
In [ ]: db = pd.DataFrame(data)
```

```
In [ ]: db
```

```
Out[8]:
```

|   | Name   | Age | Roll_No |
|---|--------|-----|---------|
| 0 | Ajay   | 23  | 1       |
| 1 | Suresh | 25  | 2       |
| 2 | Ram    | 28  | 3       |
| 3 | Mohan  | 32  | 4       |

```
In [ ]: s = pd.Series([1,2,3,np.nan,8,9])
```

```
In [ ]: s
```

```
Out[11]: 0    1.0
1    2.0
2    3.0
3    NaN
```

# 13. Data visualization

Jupyter Data\_Visualization (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

## Data visualization

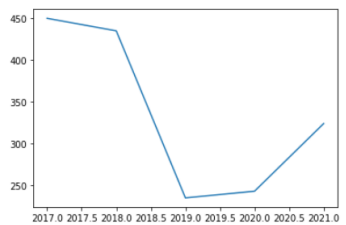
In [2]: `!pip install matplotlib`

Requirement already satisfied: matplotlib in /home/user/snap/jupyter/common/lib/python3.7/site-packages (3.3.4)  
Requirement already satisfied: pillow>=6.2.0 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from matplotlib) (8.1.2)  
Requirement already satisfied: python-dateutil>=2.1 in /snap/jupyter/6/lib/python3.7/site-packages (from matplotlib) (2.8.0)  
Requirement already satisfied: kiwisolver>=1.0.1 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from matplotlib) (1.3.1)  
Requirement already satisfied: cycler>=0.10 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from matplotlib) (0.10.0)  
Requirement already satisfied: numpy>=1.15 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from matplotlib) (1.20.1)  
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from matplotlib) (2.4.7)  
Requirement already satisfied: six>=1.5 in /snap/jupyter/6/lib/python3.7/site-packages (from python-dateutil>=2.1->matplotlib) (1.12.0)

In [5]: `#Line Chart`

```
x = [2017,2018,2019,2020,2021]
y = [450,435,235,243,324]
plt.plot(x,y)
```

Out[5]: `[<matplotlib.lines.Line2D at 0x7f3728a3bfd0>]`



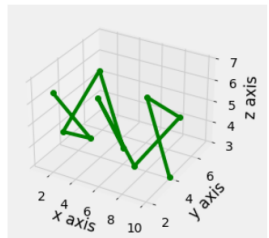
In [6]: `years = [1950,1955,1960,1965,1970,1975,1980,1985,1990,1995,2000,2005,2010,2015]`  
`pops = [2.3,4.5,2.3,1.2,5.2,1.2,4.2,1.3,2.3,1.8,1.9,2.1,1.6,2.1]`  
`plt.plot(years,pops,color='red')`

Jupyter Data\_Visualization (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3


```
ax1.set_ylabel('y axis')
ax1.set_xlabel('z axis')
```

Out[43]: `Text(0.5, 0, 'z axis')`

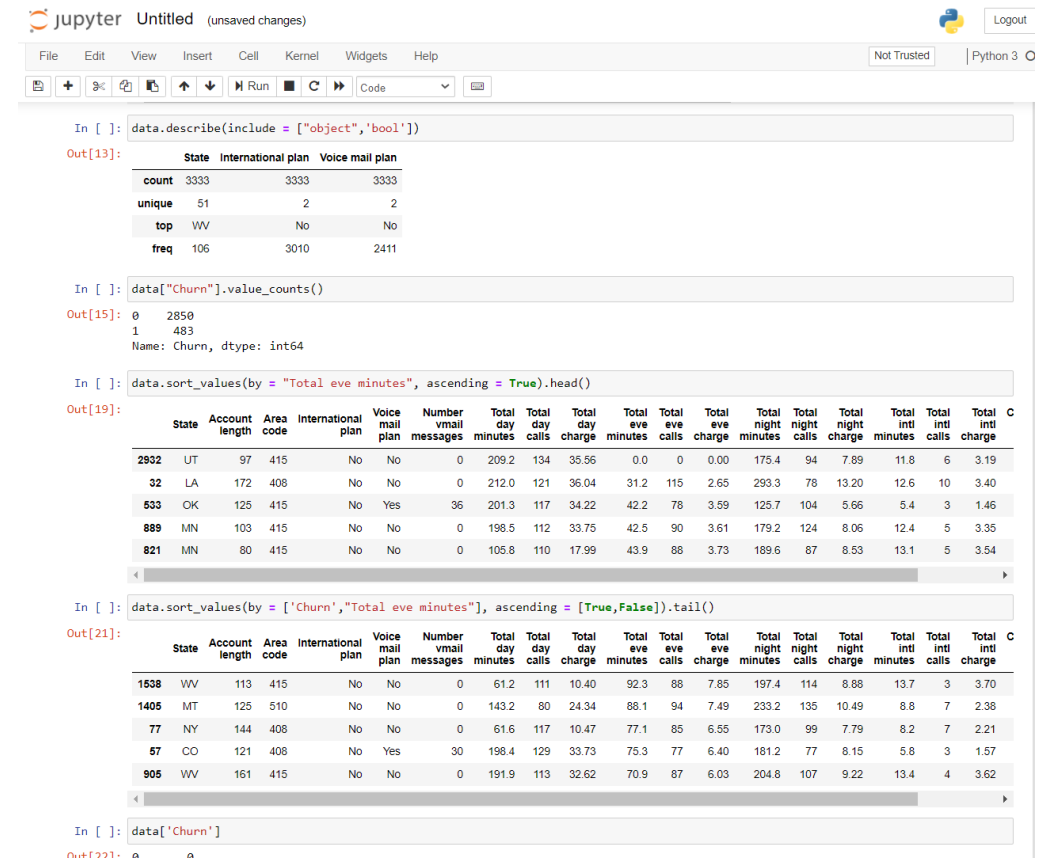
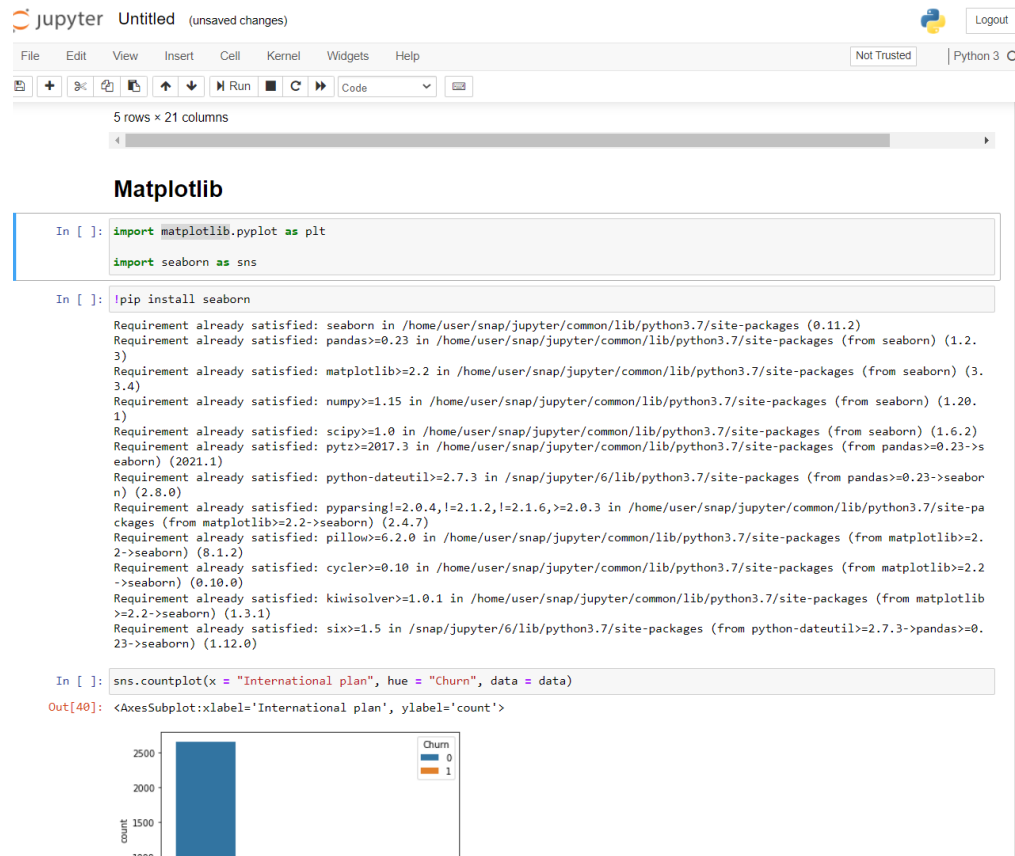


In [44]: `from mpl_toolkits.mplot3d import axes3d`  
`import matplotlib.pyplot as plt`  
`from matplotlib import style`  
`style.use('fivethirtyeight')`  
`fig = plt.figure()`  
`ax1 = fig.add_subplot(111,projection = '3d')`  
`x = [1,2,3,4,5,6,7,8,9,10]`  
`y = [3,5,2,4,5,2,4,7,3,4]`  
`z = [6,3,5,7,3,7,3,4,7,3]`  
`ax1.scatter(x,y,z,c = 'g',marker = 'o')`  
`ax1.set_xlabel('x axis')`  
`ax1.set_ylabel('y axis')`  
`ax1.set_zlabel('z axis')`

Out[44]: `Text(0.5, 0, 'z axis')`



# 14. Matplotlib



# Conclusion

- From this internship, I was able to learn basic Machine learning and what goes on behind Artificial intelligent technologies.
- The basic being the programming language used “Python”, it was essential that it was understood well. We were taught the language with depth, and it was very helpful to further our understanding with Machine Learning.
- This gave me an opportunity to discover and learn many new things and apply them to get a significant outcome. Although my technical skills in this area are still lacking with enough time and practice, I would be able to improve upon these.
- Overall, this internship has been an excellent and rewarding experience.