INTERNSHIP REPORT: DLITHE –Machine Learning with Python

Submitted by:- Achal Rai

 3rd year B.E. Computer Science & Engineering

NMAM Institute of Technology
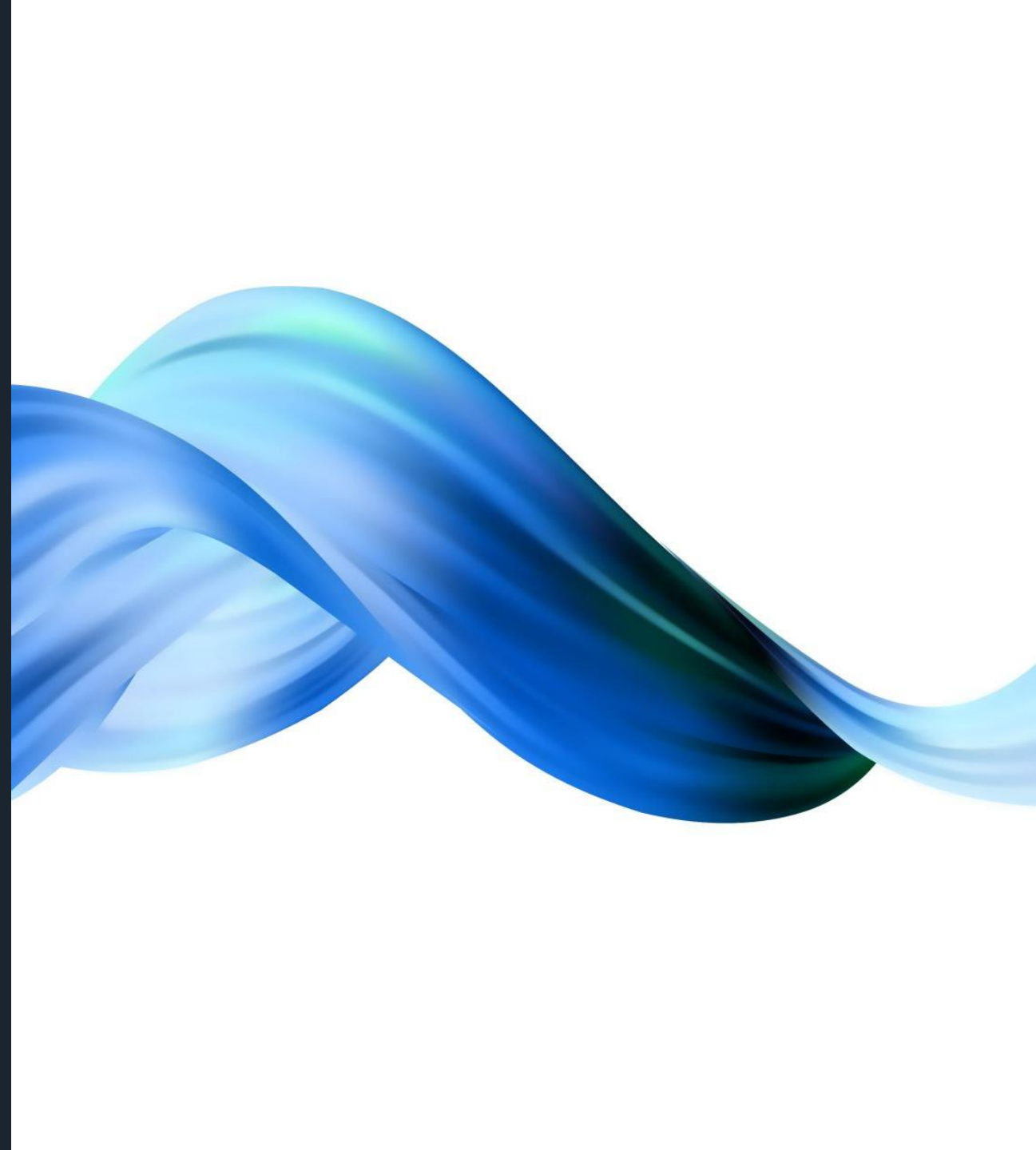
Table of Contents

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
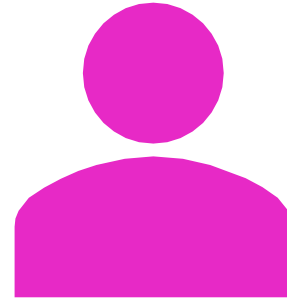
➢ DLithe is a consultancy service company from Bangalore.

➢ 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 <u>artificial intelligence (AI)</u> 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)

File | Edit | View | Insert | Cell | Kernel | Widgets | Help

Not Trusted | Python 3 ○

Markdown

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)

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

Markdown

```
In [ ]: false = 2
        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

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

Not Trusted | Python 3 ○

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

While Loop

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

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

         print()
```

```
All okay
All okay
All okay
```

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

```
4
3
2
1
```

```python
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

Code

Statement

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

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

```
In [ ]:  print(s)

         78
```

```
In [ ]:  s = "oeifhoifhwoiejgpiwejgpijgoiwrgwpigjpwijgldpiwjrouehfpiwrhj[owjfoqwfwpiegj[oehkpotfjioehqiehjowhjpwogjwpi]]"
```

```
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

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

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

Markdown

Not Trusted   Python 3

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 = 'sdfbiqb8349dhjqr83yqdho33r'
        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

Logout

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

Not Trusted | Python 3 ○

Code

Out[4]: [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

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

Not Trusted | Python 3 ○

Code

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, 'Color': 'red', 'Name': 'Ajay'}

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

```
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

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Code

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

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

Not Trusted | Python 3 ○

```
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

13. Data visualization

File | Edit | View | Insert | Cell | Kernel | Widgets | Help

Not Trusted | Python 3 ○

Markdown

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 matplotli
b) (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) (1.2
0.1)
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-pa
ckages (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]:
```python
#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>]`
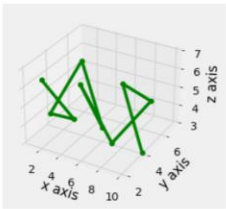
In [6]:
```python
years = [1950,1955,1960,1965,1970,1975,1980,1985,1990,1995,2000,2005,2010,2015]
pops = [2.3,4.5,2.3,1.2,5.2,1.2,4.2,1.3,2.3,1.8,1.9,2.1,1.6,2.1]
plt.plot(years,pops,color = 'red')
```
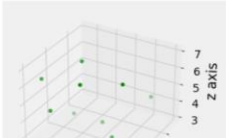
```python
ax1.set_ylabel('y axis')
ax1.set_zlabel('z axis')
```

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

In [44]:
```python
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

File | Edit | View | Insert | Cell | Kernel | Widgets | Help

Not Trusted | Python 3

5 rows × 21 columns

Matplotlib

```python
import matplotlib.pyplot as plt

import seaborn as sns
```

```python
!pip install seaborn
```

```
Requirement already satisfied: seaborn in /home/user/snap/jupyter/common/lib/python3.7/site-packages (0.11.2)
Requirement already satisfied: pandas>=0.23 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from seaborn) (1.2.3)
Requirement already satisfied: matplotlib>=2.2 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from seaborn) (3.3.4)
Requirement already satisfied: numpy>=1.15 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from seaborn) (1.20.1)
Requirement already satisfied: scipy>=1.0 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from seaborn) (1.6.2)
Requirement already satisfied: pytz>=2017.3 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from pandas>=0.23->seaborn) (2021.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /snap/jupyter/6/lib/python3.7/site-packages (from pandas>=0.23->seaborn) (2.8.0)
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>=2.2->seaborn) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (8.1.2)
Requirement already satisfied: cycler>=0.10 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/user/snap/jupyter/common/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (1.3.1)
Requirement already satisfied: six>=1.5 in /snap/jupyter/6/lib/python3.7/site-packages (from python-dateutil>=2.7.3->pandas>=0.23->seaborn) (1.12.0)
```

```python
sns.countplot(x = "International plan", hue = "Churn", data = data)
```

```
<AxesSubplot:xlabel='International plan', ylabel='count'>
```

File | Edit | View | Insert | Cell | Kernel | Widgets | Help

Not Trusted | Python 3

```python
data.describe(include = ["object","bool"])
```

	State	International plan	Voice mail plan
count	3333	3333	3333
unique	51	2	2
top	WV	No	No
freq	106	3010	2411

```python
data["Churn"].value_counts()
```

```
0    2850
1     483
Name: Churn, dtype: int64
```

```python
data.sort_values(by = "Total eve minutes", ascending = True).head()
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls	Total intl charge
2932	UT	97	415	No	No	0	209.2	134	35.56	0.0	0	0.00	175.4	94	7.89	11.8	6	3.19
32	LA	172	408	No	No	0	212.0	121	36.04	31.2	115	2.65	293.3	78	13.20	12.6	10	3.40
533	OK	125	415	No	Yes	36	201.3	117	34.22	42.2	78	3.59	125.7	104	5.66	5.4	3	1.46
889	MN	103	415	No	No	0	198.5	112	33.75	42.5	90	3.61	179.2	124	8.06	12.4	5	3.35
821	MN	80	415	No	No	0	105.8	110	17.99	43.9	88	3.73	189.6	87	8.53	13.1	5	3.54

```python
data.sort_values(by = ['Churn',"Total eve minutes"], ascending = [True,False]).tail()
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls	Total intl charge
1538	WV	113	415	No	No	0	61.2	111	10.40	92.3	88	7.85	197.4	114	8.88	13.7	3	3.70
1405	MT	125	510	No	No	0	143.2	80	24.34	88.1	94	7.49	233.2	135	10.49	8.8	7	2.38
77	NY	144	408	No	No	0	61.6	117	10.47	77.1	85	6.55	173.0	99	7.79	8.2	7	2.21
57	CO	121	408	No	Yes	30	198.4	129	33.73	75.3	77	6.40	181.2	77	8.15	5.8	3	1.57
905	WV	161	415	No	No	0	191.9	113	32.62	70.9	87	6.03	204.8	107	9.22	13.4	4	3.62

```python
data['Churn']
```

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.