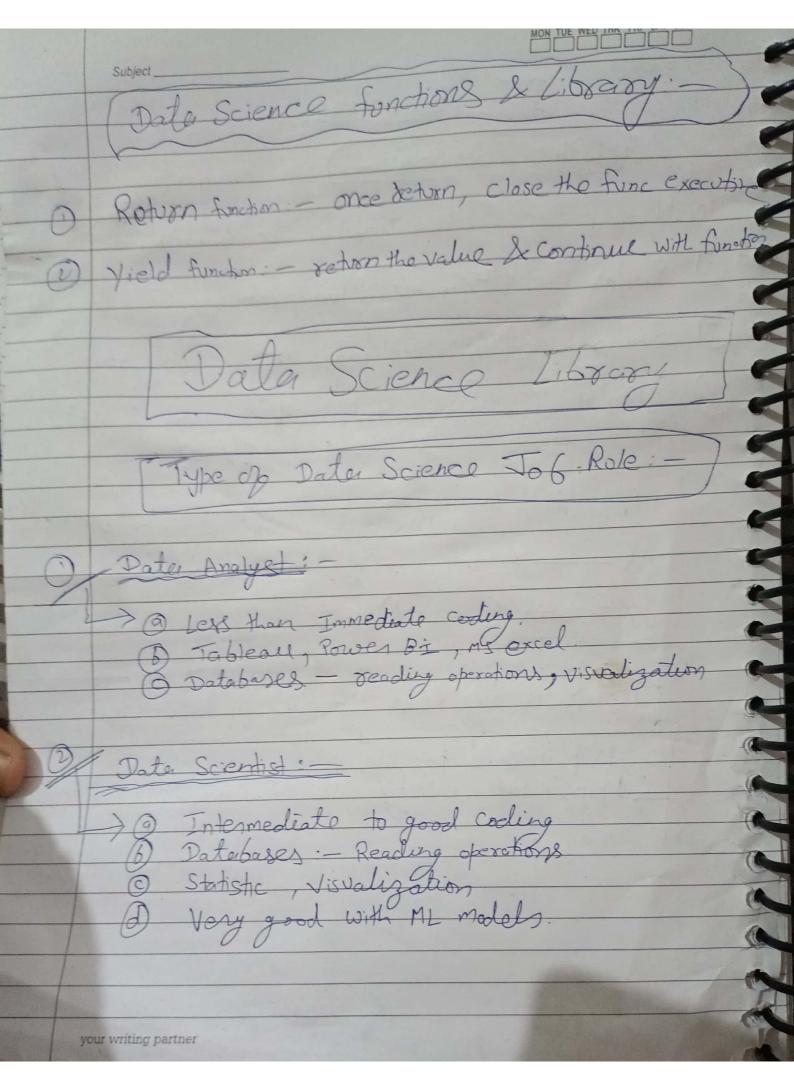Subject _____

Data Science

Python Libraries :—

① Numpy for Mathematical Calculations.

② Pandas for Dataframe Manipulations.

③ Seaborn and Matplotlib for Data Visualizations.

④ Set up the Background Style with "Fivethirtyeight". It is a name for the Background effect.

⑤ Ipywidgets Library for Interactive analysis.

———— X ———— X ———— X ———— X ————

~~X = char ("Ambarish"):~~
~~for i in range~~

To Run any new Library in Jupiter Library.

→ !Pip install pandas
→ !pip install numpy

Subject _____

Data Science functions & Library :-

① Return function :- once return, close the func execution

② yield function :- return the value & continue with function

Data Science Library

Type of Data Science Job Role :-

① **Data Analyst :-**

→ ⓐ Less than Immediate coding.
ⓑ Tableau, Power Bi, MS excel.
ⓒ Databases — reading operations, visualization

② **Data Scientist :-**

→ ⓐ Intermediate to good coding
ⓑ Databases :- Reading operations
ⓒ Statistic, Visualization
ⓓ Very good with ML models.

③ Machine learning Engineer :-

→ ⓐ Data Scientist (visualisation)
ⓑ ML Skills are very good.

④ Data Engineer :-

→ ⓐ Data Processing at Database Level.
ⓑ Bringing Data from customer to product, Data Pipelines
ⓒ SQL, Apache spark (Data processing), Airflow (Pipelines).

✗———✗———✗———✗———✗———

⊛ Different between Lists & Tuples :-

Lists	Tuples
① Lists are enclosed in Square Brackets [ ].	Tuples are enclosed in Parentheses ().
② Lists are changable (Mutable).	Tuples are unchangable (immutable)

, Operations on Lists :—

① ✓ append ()

→ append () method adds an element at the end of the list.

┌─────────────┐
│ for eg :—   │
└─────────────┘

define a list called "s"

S = ["apple", "banana", "cherry"]

using append method
S.append ("cauliflower")
print ("s")

┌────────┐
│ output │
└────────┘

["apple", "banana", "cherry", "cauliflower"]

② Insert ()

→ Insert () method adds an element at the specified index.

┌──────────────┐
│ for example │
└──────────────┘

using insert () method
s.insert (0, "bringal")
print (s)

┌────────┐
│ output │
└────────┘
["bringal", "apple", "banana", "cherry", "cauliflower"]

③ Remove ()

→ remove () method removes the item with the specified value.

[Example :-]

\# using remove () method :-

S.remove ("brinjal")
print (s)

[Output]

⇒ ['apple', 'banana', 'cherry', 'cauliflower']

④ Pop () :-

→ Pop () method removes the element at the specified index (or last item if the index is not specified).

[Example :-]

\# using pop() method :-

S.pop()
print (s)

[output]
['apple', 'banana', 'cherry']

5 / Copy () :-

→ Copy() method returns a copy of the list.

example :-

```
# using copy() method
Scopy = S.copy()
print (Scopy)
```

Output

['apple', 'banana', 'cherry'].

6 Clear() :-

→ clear() method removes all elements from the list.

Example:-

```
# using clear() method

BScopy.clear()
print (Scopy)
```

output

→ [ ]

empty list

Subject _____

(7) reverse ( ) :-

→ Reverse ( ) method reverses the order of the list.

[Example :-]

using Reverse ( ) method

S. reverse ( )
Print (S)

[Output]

= ['cherry', 'banana', 'apple'].

(8) Count ( ) :-

→ Count ( ) method returns the no. of elements with the specified value.

[Example :-]

→ # Using count ( ) method :-

S. count (' apple ')

[Output :-]
→ 1

9) Extend() :-

→ extend() method adds the elements of the list to the current list.

Example :-

→ # using extend() method

```
S1 = [3, 8]
S.extend(S1)
print S
```

Output

→ ['cherry', 'banana', 'apple', 3, 8]

10) Sort() :-

→ sort() method sorts the list.

Example :-

→ # using sort() method

```
S = ["apple", "banana", "cherry"]
S.sort()
print(s)
```

Output

→ ["apple", "banana", "cherry"]

Another example of sort () method :-

```
a = [4,7,2,9,0]
a.sort()
print(a)
```

Output

→ [0,2,4,7,9] //

━━━━✖━━━━✖━━━━✖━━━━✖━━━━✖━━━━

Operations on Tuple

① Count () :-

→ Count() method returns the number of times a specified value occur in a tuple.

Example :-

→ # using count() method :-
```
t = ("apple", "banana", "cherry", "banana")
t.count("banana")
```

Output

→ 2

② Index():-

→ Index() method searches the tuple for a specified value and Returns the position of where it was found.

Example:-

→ # using index() method :-
    t. index ("cherry")

output

    → ②

③ len():-

→ len() method returns the length of the tuple

Example:

→ # using len() method
    len (t)

output

    → 4

④ " + "operator :-

→ " + "operator is used to join two tuples.

example :-

↳ # using '+' operator to join tuples :-

$$U_1 = (2,5,7,6)$$
$$U_2 = (1,2,4,5)$$

$$U_1 + U_2$$

output :-

↳ (2,5,7,6,1,2,4,5) //