Buildable ML/DL Fellowship

Week#01 Report

Name: Muhammad Abdullah Shariq

Date: 24/08/2025

Q2)

1.

```
week1_assignment.py > ...

print("/n Q2-Mutable Vs Immutable/n")
#1.
tup=(1,2,3)
tup[0]=10
# This gives an error bcz tuple is immutable(cannot be changed)

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"
/n Q2-Mutable Vs Immutable/n
Traceback (most recent call last):
  File "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py", line 5, in <module>
    tup[0]=10
    ~~~^^^
TypeError: 'tuple' object does not support item assignment
PS F:\Buildables ML-DL Fellowship\week1>
```

Tuple is Immutable (values cannot be changed)

2.

```
week1_assignment.py > ...

# print("/n Q2-Mutable Vs Immutable/n")
# #1.
# tup=(1,2,3)
# tup[0]=10
# # This gives an error bcz tuple is immutable(cannot be changed)

#2.
list=[4,5,6]
list[0]=20
print(list)
# # This works fine as List is Mutable(can be changed)

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"
[20, 5, 6]
PS F:\Buildables ML-DL Fellowship\week1>
```

List is Mutable (values can be changed)

3.

```
week1_assignment.py > ...
# list[0]=20
# print(list)
# # This works fine as List is Mutable(can be changed)

# #3.
dict = {"name": "Abdullah", "age": 21}
dict["age"]=50
print(dict)
# #This works fine as Keys's value can be updated

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"
{'name': 'Abdullah', 'age': 50}
PS F:\Buildables ML-DL Fellowship\week1>
```

Dictionary Key's values can be changed

4.

```
week1_assignment.py > ...
# print(dict)
# #This works fine as Key's value can be updated

#4.
tuple_list=([1,2],[3,4])
tuple_list[0][0]=50
print(tuple_list)
# This works fine because list inside the tuple is mutable and can be modified

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"
([50, 2], [3, 4])
PS F:\Buildables ML-DL Fellowship\week1>
```

The tuple is immutable, but it can hold mutable objects like lists, which can still be modified.

<u>Mutable vs Immutable in Python</u>

Immutable objects cannot be changed after creation. Eg: tuple, string, int, float.

In our code, the Tuple itself is immutable, hence we cannot reassign tup[0]=10.

Mutable objects can be modified after creation. Eg: list, dict, set.

In our code, the List is mutable, hence we are able to change list[0]=20.

Q3)

week1_assignment.py > ...

```python
print("/nQ3 User Information Dictionary (Validation + Logic)/n")

user={}

while True:

    name=input("Enter your name:")
    age=int(input("Enter your age:"))

    if age<=0 or age>=100:
        print("Invalid age")
        continue

    email=input("Enter your email:")

    if "@" not in email or "." not in email or email[0]in"!@#." or email[-1]in"!@#.":
        print("Invalid email")
        continue

    fav=int(input("Enter favorite number(1-100):"))
    if not (1<=fav<=100):
        print("Invalid number")
        continue
    break
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"
/nQ3 User Information Dictionary (Validation + Logic)/n
Enter your name:Abdullah
Enter your age:110
Invalid age
Enter your name:Abdullah Shariq
Enter your age:22
Enter your email:abdullah.shariq05@gmail.com
Enter favorite number(1-100):5
WelcomeAbdullah Shariq! Your account has been registered with email abdullah.shariq05@gmail.com.
PS F:\Buildables ML-DL Fellowship\week1>
```

Q4)

week1_assignment.py ✕ test.py

week1_assignment.py > ...

```python
print("/nQ4 Cinema Ticketing System/n")

def calculate_ticket_price(age,is_student,is_weekend):
    if age<0 or age>120:
        print("Invalid age")

    if age<12:
        price=5
    elif age<=17:
        price=8
    elif age<=59:
        price=12
    else:
        price=6

    if is_student and age>12:
        price*=0.8
    if is_weekend:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"
/nQ4 Cinema Ticketing System/n
Number of customers:4
Age:-5
Student (yes/no): yes
Weekend (yes/no): yes
Invalid age
Age:10
Student (yes/no): no
Weekend (yes/no): no
Age:15
Student (yes/no): no
Weekend (yes/no): no
Age:16
Student (yes/no): yes
Weekend (yes/no): no
Total Revenue: $26.4
PS F:\Buildables ML-DL Fellowship\week1>
```

Q5)

week1_assignment.py test.py

week1_assignment.py > ...

```python
print("\nQ5 Weather Alert System\n")

def weather_alert(temp,condition):
    alert="Normal weather conditions"

    if temp<0 and condition=="snowy":
        alert="Heavy snow alert,Stay Indoors"
    elif temp>35 and condition=="sunny":
        alert="Heatwave warning,Stay Hydrated"
    elif condition=="rainy" and temp<15:
        alert="Cold rain alert,Wear warm clothes"

    fah=temp*(9/5)+32
    k=temp+273.15

    return f"{alert} ({temp}°C / {fah:.1f}°F / {k:.2f}K)"

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"

Q5 Weather Alert System

Enter temperature in Celsius:-5
Enter weather condition (sunny/rainy/snowy/etc):snowy
Heavy snow alert,Stay Indoors (-5.0°C / 23.0°F / 268.15K)
PS F:\Buildables ML-DL Fellowship\week1>
```

Q6)

week1_assignment.py test.py

week1_assignment.py > ...

```python
print("\nQ6 Sales Analytics (Max, Min & Median)\n")

def analyze_sales(sales):
    sales.sort()
    n=len(sales)
    median=sales[n//2] if n%2!=0 else(sales[n//2-1]+sales[n//2])/2
    return max(sales),min(sales),median

sales=[]
while len(sales)<5:
    sales.append(float(input("Enter daily sale:")))

max_sale,min_sale,median_sale=analyze_sales(sales)

print(f"Highest sales day: {max_sale}")
print(f"Lowest sales day: {min_sale}")
print(f"Median sales: {median_sale}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"

Q6 Sales Analytics (Max, Min & Median)

Enter daily sale:100
Enter daily sale:200
Enter daily sale:300
Enter daily sale:400
Enter daily sale:500
Highest sales day: 500.0
Lowest sales day: 100.0
Median sales: 300.0
PS F:\Buildables ML-DL Fellowship\week1>
```

Q7)

week1_assignment.py test.py

week1_assignment.py > update_inventory

```python
print("\nQ7 E-commerce Inventory Management\n")

def update_inventory(inv,item,qty):
    if item not in inv:
        inv[item]=0

    inv[item]=max(inv[item]+qty,0)
    return inv

inventory={"apple":10,"banana":5,"milk":20,"bread":15,"egg":30}
for _ in range(3):
    item=input("Item:")
    qty=int(input("Quantity:"))
    if qty>inventory.get(item,0):
        print(f"Not enough stock for {item}")
    else:
        update_inventory(inventory,item,-qty)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
PS F:\Buildables ML-DL Fellowship\week1> python -u "f:\Buildables ML-DL Fellowship\week1\week1_assignment.py"

Q7 E-commerce Inventory Management

Item:apple
Quantity:3
Item:banana
Quantity:2
Item:milk
Quantity:5
Updated inventory: {'apple': 7, 'banana': 3, 'milk': 15, 'bread': 15, 'egg': 30}
Most stocked: egg
Least stocked: banana
PS F:\Buildables ML-DL Fellowship\week1>
```

Q9)

1. Difference between AI, Machine Learning, Deep Learning, and Data Science

- **Artificial Intelligence (AI):**
  AI is the broad field where machines are designed to mimic human intelligence.

  Example: Chatbots like Siri or Google Assistant that understand and respond to questions.

- **Machine Learning (ML):**
  ML is a subset of AI where systems **learn patterns from data** to make predictions or decisions without being explicitly programmed.

  Example: Email spam filters that learn to classify emails as spam or not.

- **Deep Learning (DL):**
  DL is a subset of ML that uses **neural networks with many layers** to model complex patterns.

  Example: Self-driving car image recognition to detect pedestrians and traffic signs.

- **Data Science:**
  Data Science is the practice of **extracting insights from data** using statistics, ML, and visualization techniques.

  Example: Netflix recommending movies based on your watch history.

2. Mutable vs Immutable Data Types

- **Mutable:** Can be changed after creation.

  Example: Lists, dictionaries, sets.

  You can modify, add, or remove elements without creating a new object.

- **Immutable:** Cannot be changed after creation.

  Example: Strings, tuples, integers.

  Any modification creates a new object instead of changing the original.

3. Difference between Shallow Copy and Deep Copy

- **Shallow Copy:** Creates a new object but **references the same inner objects** as the original.

  Modifying inner objects affects both copies.

  Example: copy.copy(list_of_lists)

- **Deep Copy:** Creates a new object and **recursively copies all nested objects**, making it independent of the original.

  Modifying one copy does not affect the other.

  Example: copy.deepcopy(list_of_lists)

4. Git Branching and Its Importance

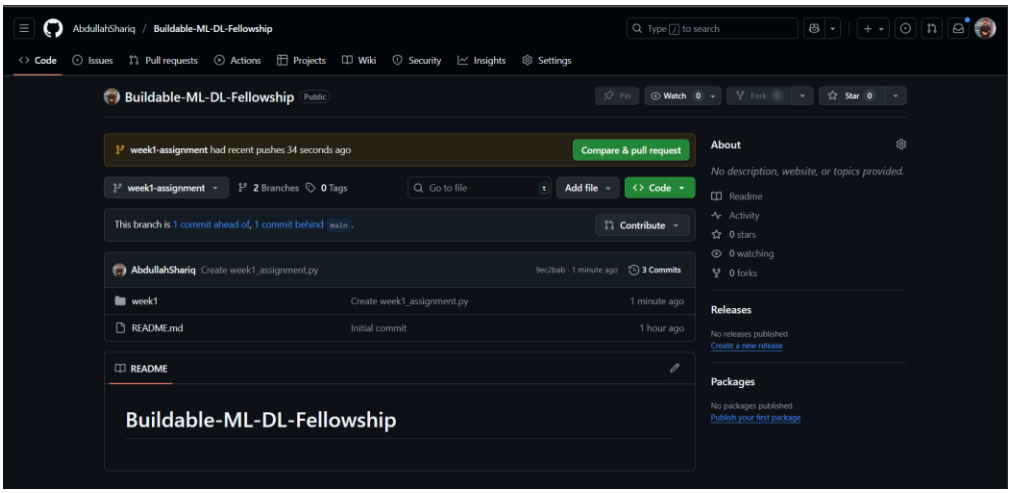
- **Git Branching:** A branch is a separate line of development in a Git repository.

  Allows multiple developers to work on features, bug fixes, or experiments **without affecting the main code**.

- **Importance:**
  1. Enables collaboration without overwriting others' work.
  2. Helps organize code by features or releases.
  3. Makes it easier to test new changes before merging into the main project.

  Example: Creating a branch feature-login to develop a login system while main stays stable.

Branch Created:



Created Pull Request: