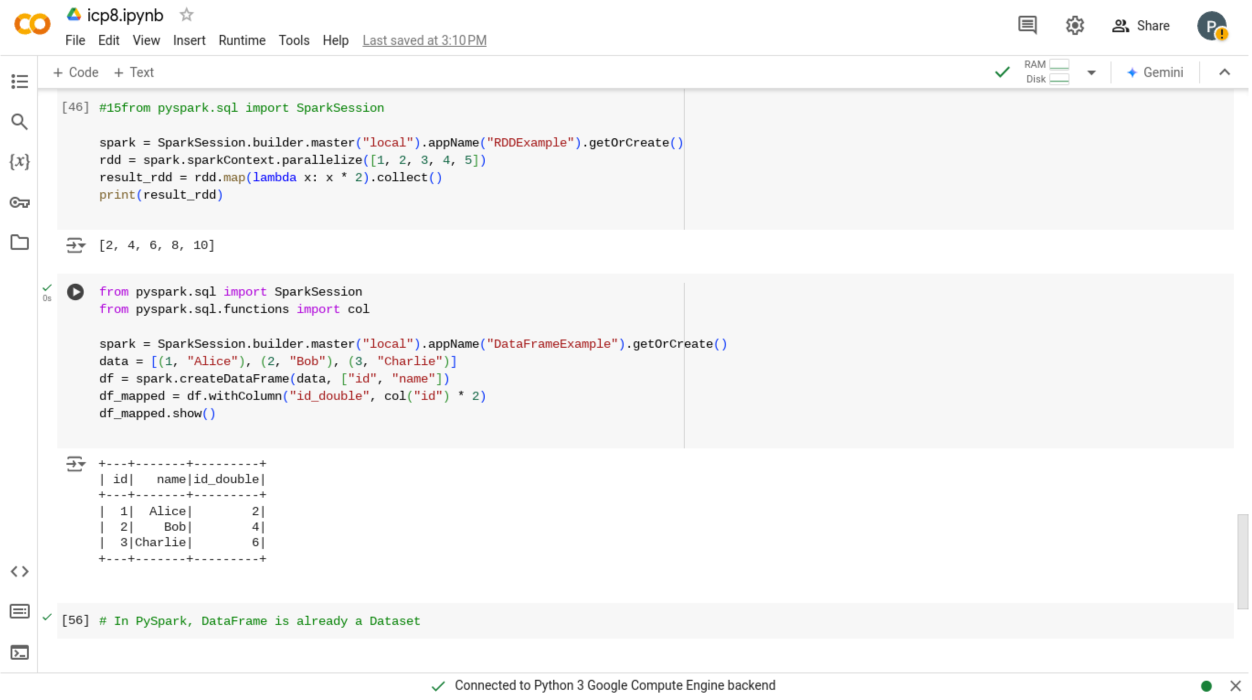
Icp8


icp8.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:10 PM
Share
+ Code + Text
RAM
Disk
Gemini
Connected to Python 3 Google Compute Engine backend

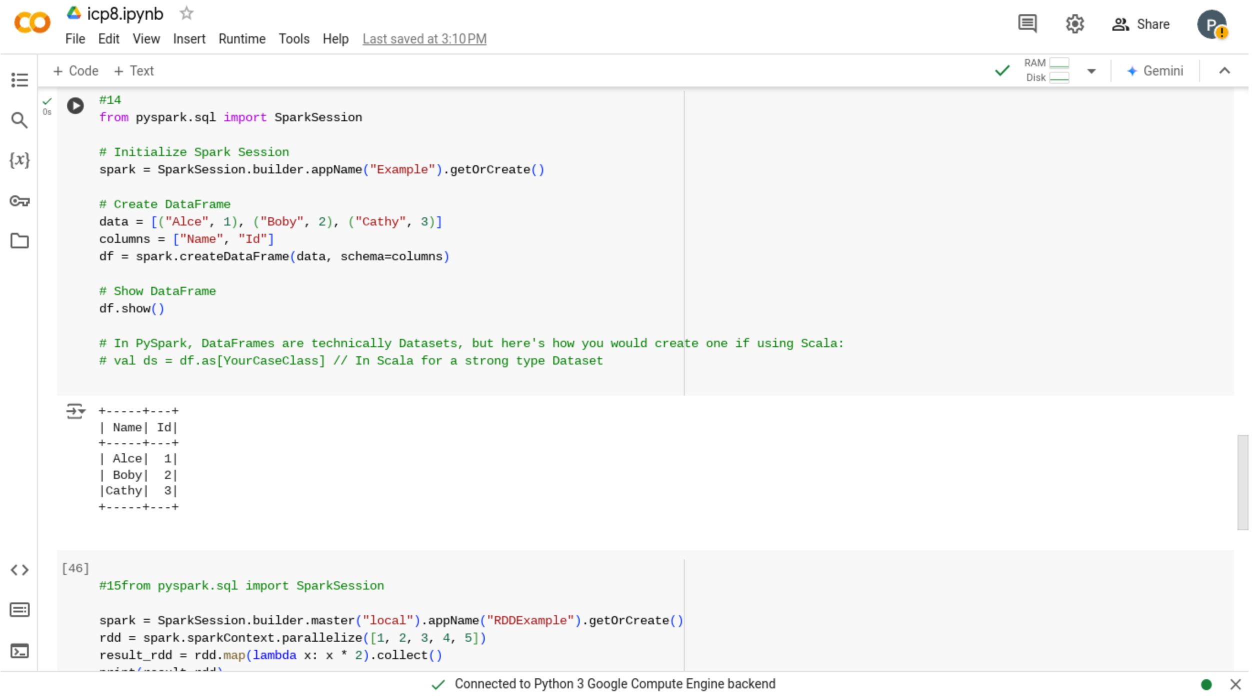
```
[46] #15from pyspark.sql import SparkSession

     spark = SparkSession.builder.master("local").appName("RDDExample").getOrCreate()
     rdd = spark.sparkContext.parallelize([1, 2, 3, 4, 5])
     result_rdd = rdd.map(lambda x: x * 2).collect()
     print(result_rdd)
```

```
[2, 4, 6, 8, 10]
```

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col

spark = SparkSession.builder.master("local").appName("DataFrameExample").getOrCreate()
data = [(1, "Alice"), (2, "Bob"), (3, "Charlie")]
df = spark.createDataFrame(data, ["id", "name"])
df_mapped = df.withColumn("id_double", col("id") * 2)
df_mapped.show()
```

```
+---+-------+---------+
| id|   name|id_double|
+---+-------+---------+
|  1|  Alice|        2|
|  2|    Bob|        4|
|  3|Charlie|        6|
+---+-------+---------+
```

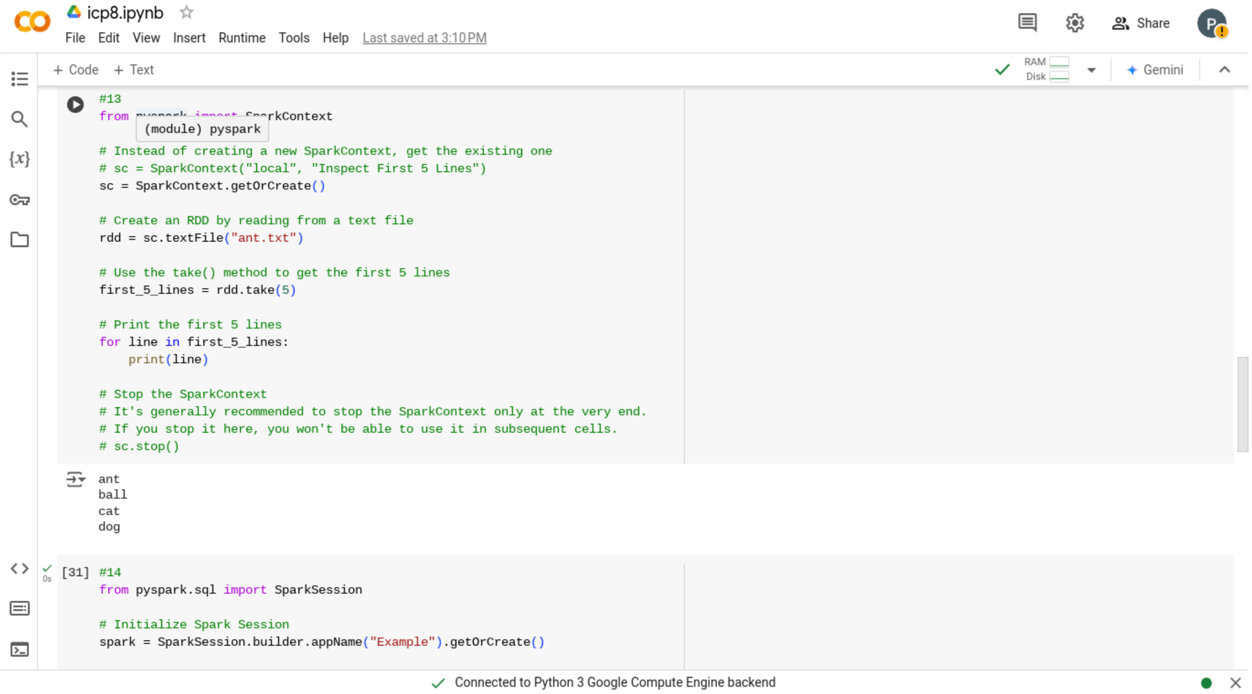
```
[56] # In PySpark, DataFrame is already a Dataset
```


icp8.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:10 PM
Share
+ Code + Text
RAM
Disk
Gemini
Connected to Python 3 Google Compute Engine backend

```
#14
from pyspark.sql import SparkSession

# Initialize Spark Session
spark = SparkSession.builder.appName("Example").getOrCreate()

# Create DataFrame
data = [("Alce", 1), ("Boby", 2), ("Cathy", 3)]
columns = ["Name", "Id"]
df = spark.createDataFrame(data, schema=columns)

# Show DataFrame
df.show()

# In PySpark, DataFrames are technically Datasets, but here's how you would create one if using Scala:
# val ds = df.as[YourCaseClass] // In Scala for a strong type Dataset
```

```
+-----+---+
| Name| Id|
+-----+---+
| Alce|  1|
| Boby|  2|
|Cathy|  3|
+-----+---+
```

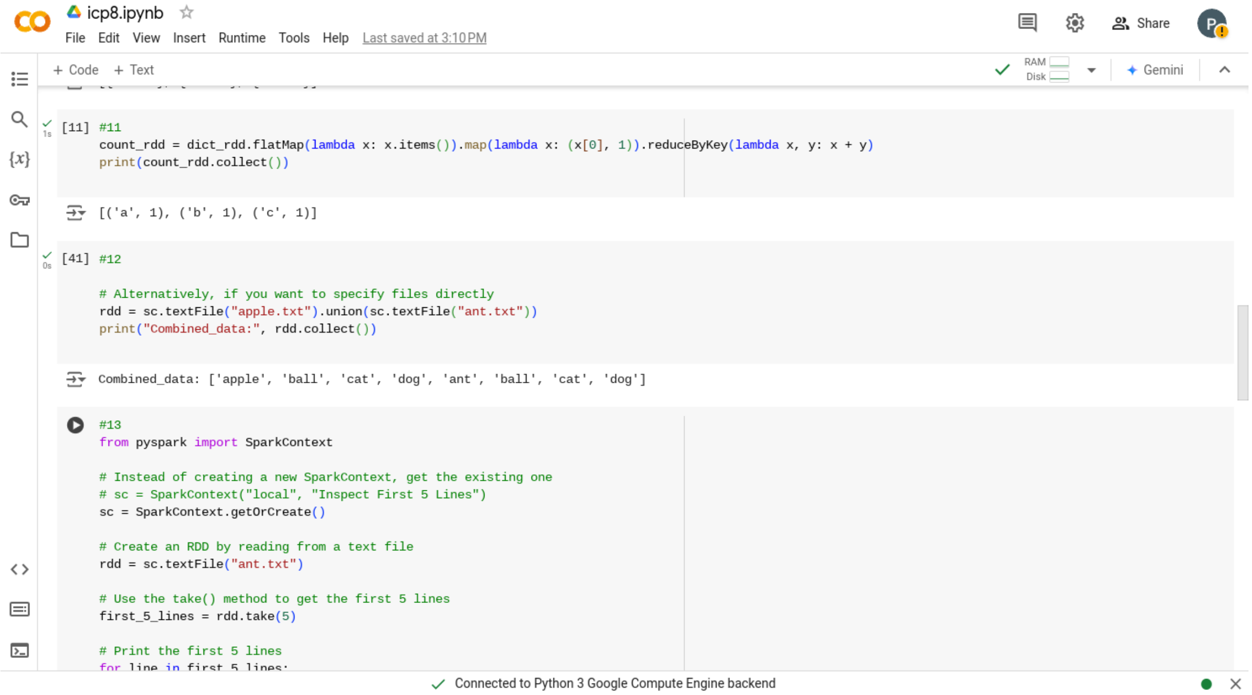
```
[46]
     #15from pyspark.sql import SparkSession

     spark = SparkSession.builder.master("local").appName("RDDExample").getOrCreate()
     rdd = spark.sparkContext.parallelize([1, 2, 3, 4, 5])
     result_rdd = rdd.map(lambda x: x * 2).collect()
     print(result_rdd)
```


icp8.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:10 PM
Share
+ Code + Text
RAM
Disk
Gemini
(module) pyspark
Connected to Python 3 Google Compute Engine backend

```
#13
from pyspark import SparkContext

# Instead of creating a new SparkContext, get the existing one
# sc = SparkContext("local", "Inspect First 5 Lines")
sc = SparkContext.getOrCreate()

# Create an RDD by reading from a text file
rdd = sc.textFile("ant.txt")

# Use the take() method to get the first 5 lines
first_5_lines = rdd.take(5)

# Print the first 5 lines
for line in first_5_lines:
    print(line)

# Stop the SparkContext
# It's generally recommended to stop the SparkContext only at the very end.
# If you stop it here, you won't be able to use it in subsequent cells.
# sc.stop()
```

```
ant
ball
cat
dog
```

```
[31] #14
from pyspark.sql import SparkSession

# Initialize Spark Session
spark = SparkSession.builder.appName("Example").getOrCreate()
```


icp8.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:10 PM
Share
+ Code + Text
RAM
Disk
Gemini
Connected to Python 3 Google Compute Engine backend

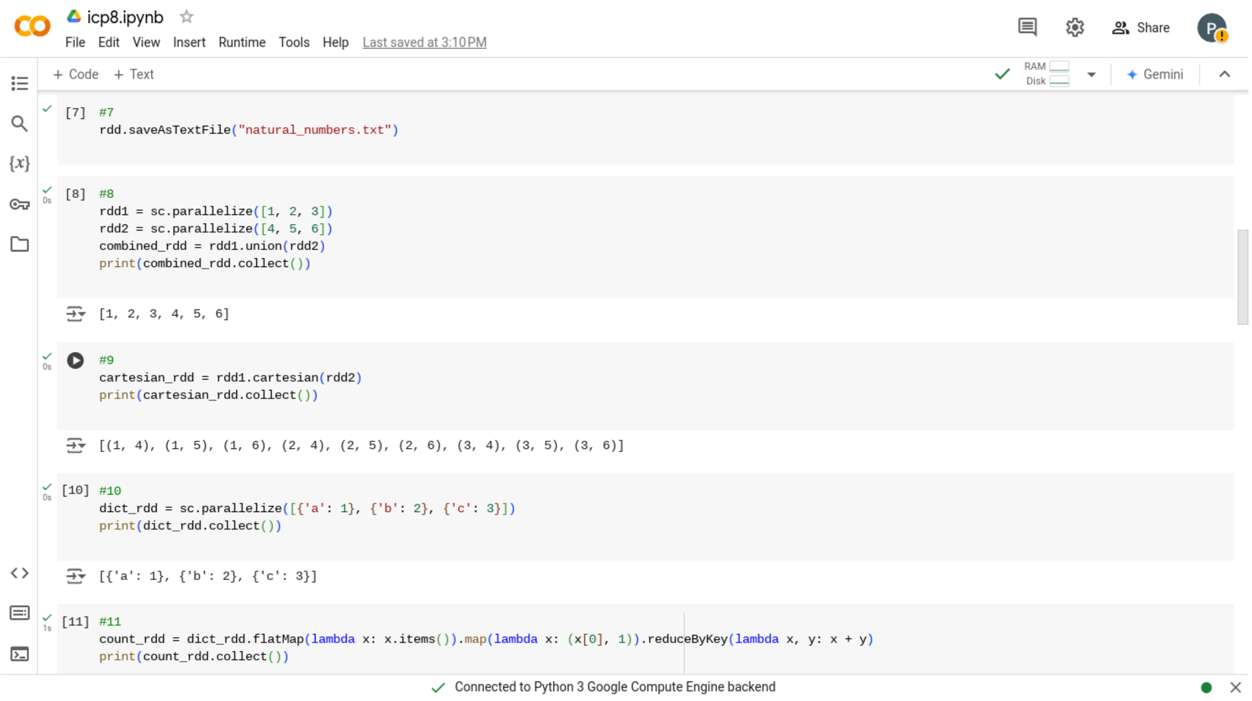
```
[11] #11
count_rdd = dict_rdd.flatMap(lambda x: x.items()).map(lambda x: (x[0], 1)).reduceByKey(lambda x, y: x + y)
print(count_rdd.collect())
```

```
[('a', 1), ('b', 1), ('c', 1)]
```

```
[41] #12

# Alternatively, if you want to specify files directly
rdd = sc.textFile("apple.txt").union(sc.textFile("ant.txt"))
print("Combined_data:", rdd.collect())
```

```
Combined_data: ['apple', 'ball', 'cat', 'dog', 'ant', 'ball', 'cat', 'dog']
```

```
#13
from pyspark import SparkContext

# Instead of creating a new SparkContext, get the existing one
# sc = SparkContext("local", "Inspect First 5 Lines")
sc = SparkContext.getOrCreate()

# Create an RDD by reading from a text file
rdd = sc.textFile("ant.txt")

# Use the take() method to get the first 5 lines
first_5_lines = rdd.take(5)

# Print the first 5 lines
for line in first_5_lines:
```


icp8.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:10 PM
Share
+ Code + Text
RAM
Disk
Gemini

```
[7] #7
    rdd.saveAsTextFile("natural_numbers.txt")
```

```
[8] #8
    rdd1 = sc.parallelize([1, 2, 3])
    rdd2 = sc.parallelize([4, 5, 6])
    combined_rdd = rdd1.union(rdd2)
    print(combined_rdd.collect())
```

```
[1, 2, 3, 4, 5, 6]
```

```
#9
cartesian_rdd = rdd1.cartesian(rdd2)
print(cartesian_rdd.collect())
```

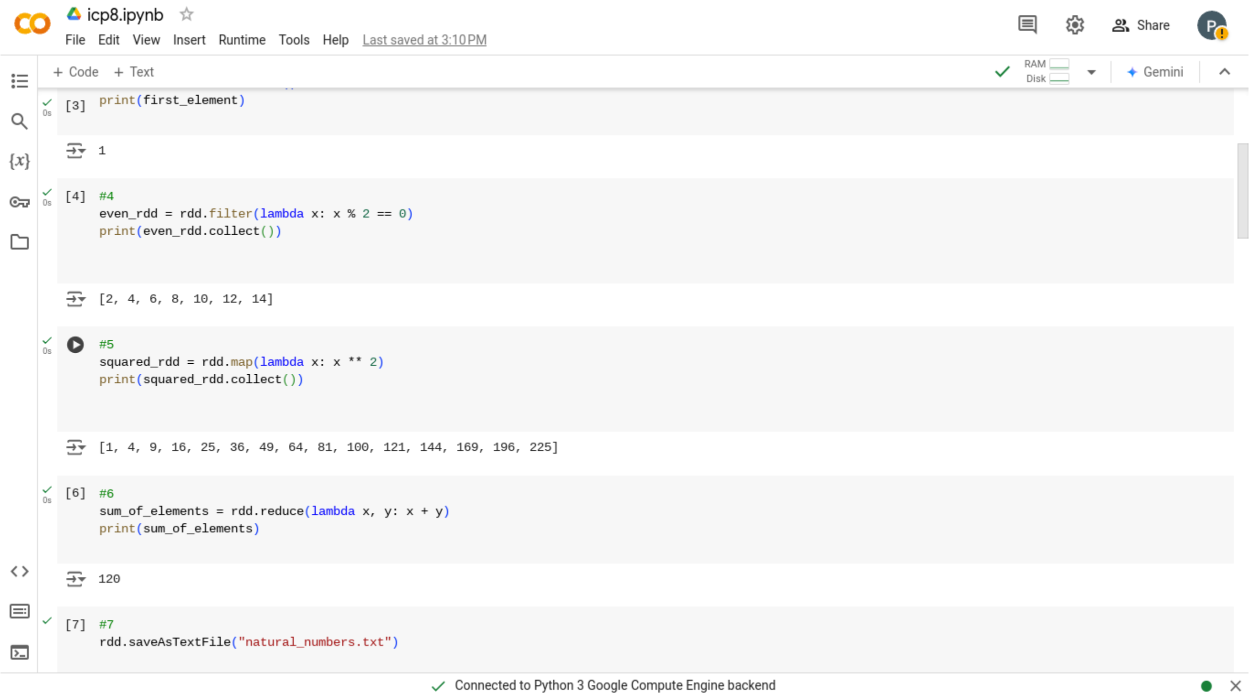
```
[(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6)]
```

```
[10] #10
     dict_rdd = sc.parallelize([{'a': 1}, {'b': 2}, {'c': 3}])
     print(dict_rdd.collect())
```

```
[{'a': 1}, {'b': 2}, {'c': 3}]
```

```
[11] #11
     count_rdd = dict_rdd.flatMap(lambda x: x.items()).map(lambda x: (x[0], 1)).reduceByKey(lambda x, y: x + y)
     print(count_rdd.collect())
```

Connected to Python 3 Google Compute Engine backend


icp8.ipynb
File Edit View Insert Runtime Tools Help Last saved at 3:10 PM
Share
+ Code + Text
RAM
Disk
Gemini

```
[3] print(first_element)
```

```
1
```

```
[4] #4
    even_rdd = rdd.filter(lambda x: x % 2 == 0)
    print(even_rdd.collect())
```

```
[2, 4, 6, 8, 10, 12, 14]
```

```
#5
squared_rdd = rdd.map(lambda x: x ** 2)
print(squared_rdd.collect())
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225]
```

```
[6] #6
    sum_of_elements = rdd.reduce(lambda x, y: x + y)
    print(sum_of_elements)
```

```
120
```

```
[7] #7
    rdd.saveAsTextFile("natural_numbers.txt")
```

Connected to Python 3 Google Compute Engine backend

Gethub: https://github.com/pavan7036/bda