GROUP 6: MEMBERS

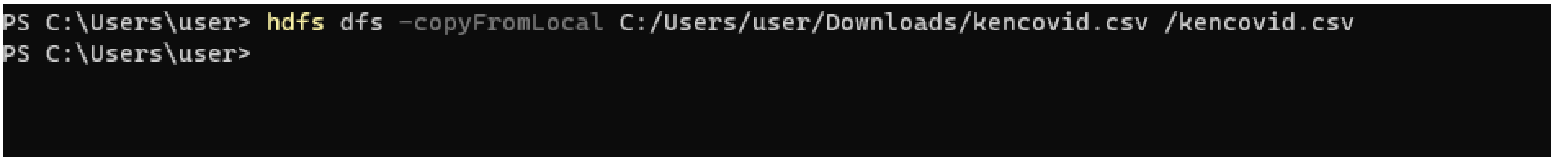
NAMES	REGNO
Wanjohi Ian Kinyua	SCT221-0409/2022
Rukia Omar	SCT221-0413/2022
Katrina Najaru	SCT221-0408/2022
Elvis Kimathi	SCT221-0504/2022
Stacy Mwaniki	SCT221-0503/2022

Question one

1. Ingest the data into Hadoop DFS Data lake

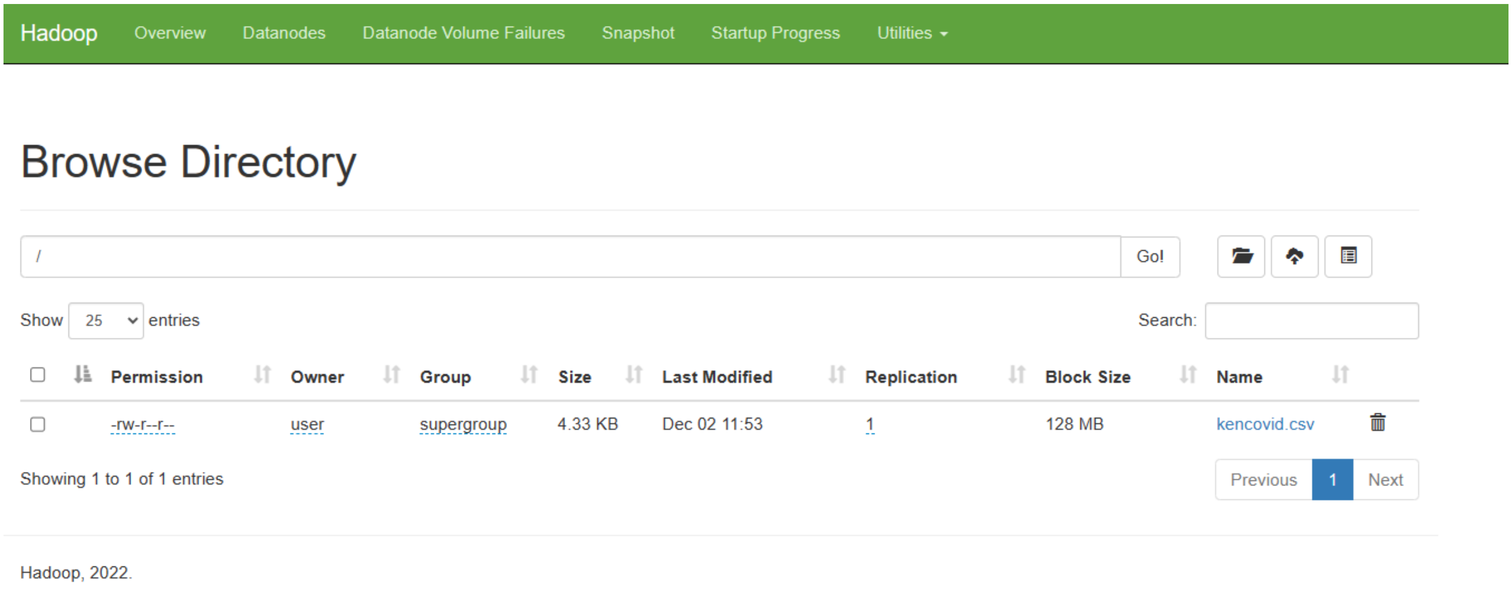
Command used :

hdfs dfs -copyFromLocal C:/Users/user/Downloads/kencovid.csv /kencovid.csv



```
PS C:\Users\user> hdfs dfs -copyFromLocal C:/Users/user/Downloads/kencovid.csv /kencovid.csv
PS C:\Users\user>
```

In Hadoop



Browse Directory

/

Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	user	supergroup	4.33 KB	Dec 02 11:53	1	128 MB	kencovid.csv

Showing 1 to 1 of 1 entries

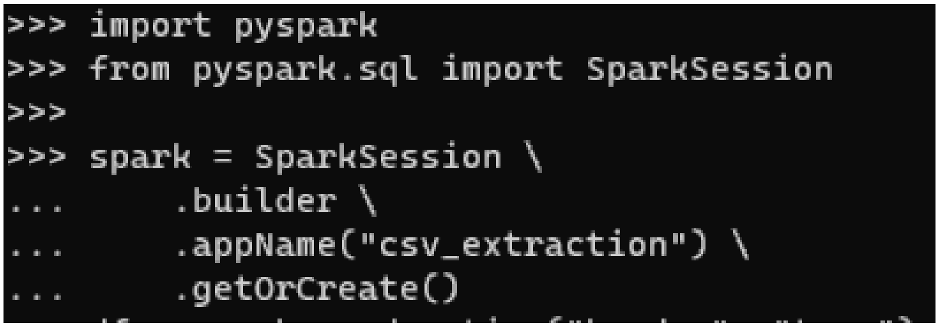
Previous 1 Next

Hadoop, 2022.

Question 2

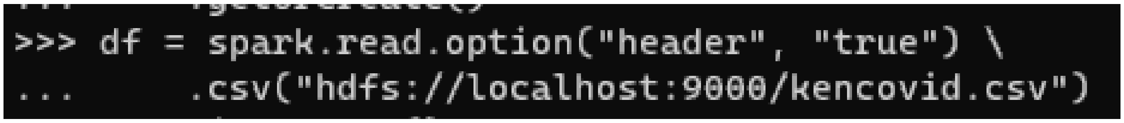
2. Use pyspark package to extract the data from the data lake

First, start a SparkSession:



```
>>> import pyspark
>>> from pyspark.sql import SparkSession
>>>
>>> spark = SparkSession \
...     .builder \
...     .appName("csv_extraction") \
...     .getOrCreate()
```

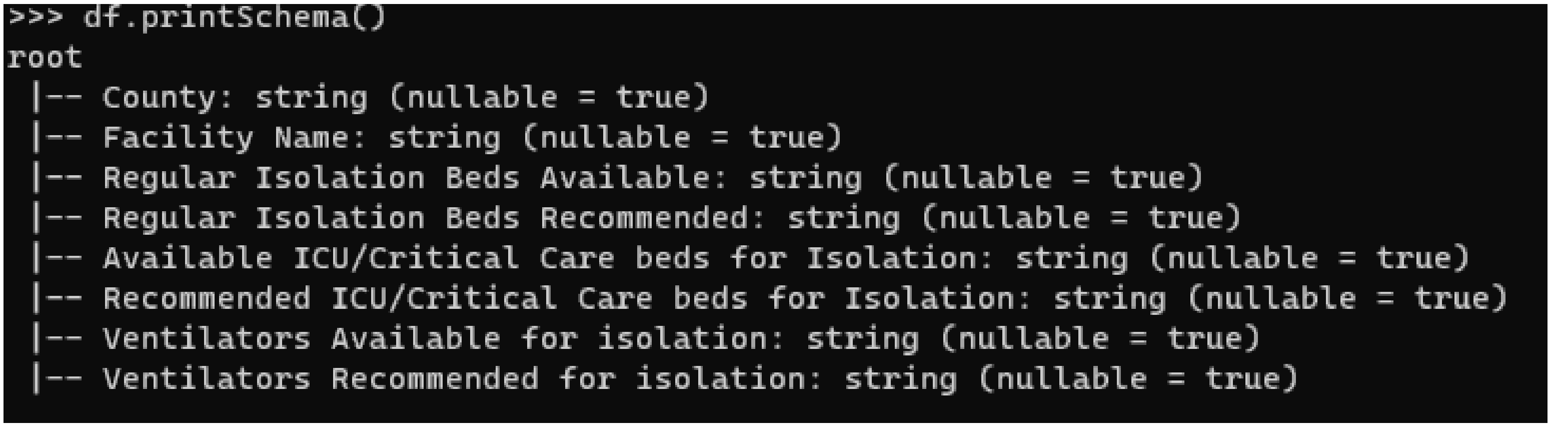
Then read the CSV file from HDFS into a DataFrame:



```
>>> df = spark.read.option("header", "true") \
...     .csv("hdfs://localhost:9000/kencovid.csv")
```

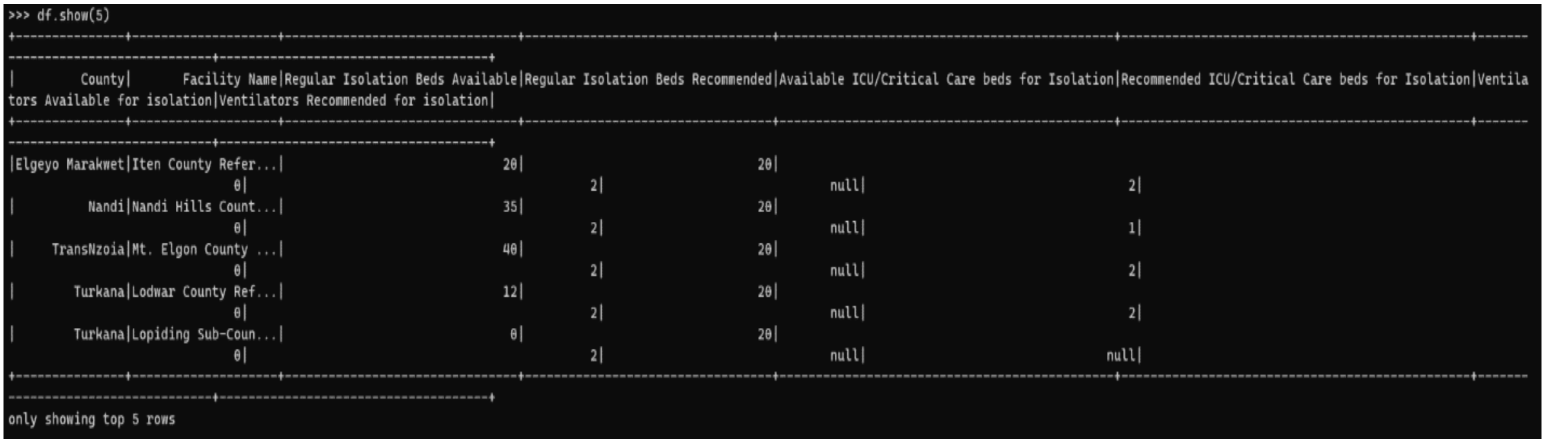
The options tell Spark to treat the first row as header.

Now the DataFrame df contains the contents of the CSV file. You can perform analysis and transformations on it. For example, to print the schema:



```
>>> df.printSchema()
root
 |-- County: string (nullable = true)
 |-- Facility Name: string (nullable = true)
 |-- Regular Isolation Beds Available: string (nullable = true)
 |-- Regular Isolation Beds Recommended: string (nullable = true)
 |-- Available ICU/Critical Care beds for Isolation: string (nullable = true)
 |-- Recommended ICU/Critical Care beds for Isolation: string (nullable = true)
 |-- Ventilators Available for isolation: string (nullable = true)
 |-- Ventilators Recommended for isolation: string (nullable = true)
```

To print the first few rows:



```
>>> df.show(5)
+---------------+--------------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-------
---------------------------+-------------------------------------+
|         County|       Facility Name|Regular Isolation Beds Available|Regular Isolation Beds Recommended|Available ICU/Critical Care beds for Isolation|Recommended ICU/Critical Care beds for Isolation|Ventila
tors Available for isolation|Ventilators Recommended for isolation|
+---------------+--------------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-------
---------------------------+-------------------------------------+
|Elgeyo Marakwet|Iten County Refer...|                              20|                                20|
                              0|                                              2|                              null|                                  2|
|          Nandi|Nandi Hills Count...|                              35|                                20|
                              0|                                              2|                              null|                                  1|
|     TransNzoia|Mt. Elgon County ...|                              40|                                20|
                              0|                                              2|                              null|                                  2|
|        Turkana|Lodwar County Ref...|                              12|                                20|
                              0|                                              2|                              null|                                  2|
|        Turkana|Lopiding Sub-Coun...|                               0|                                20|
                              0|                                              2|                              null|                               null|
+---------------+--------------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-------
---------------------------+-------------------------------------+
only showing top 5 rows
```

Question 3

3. Choose appropriate techniques to Pre- process the extracted data

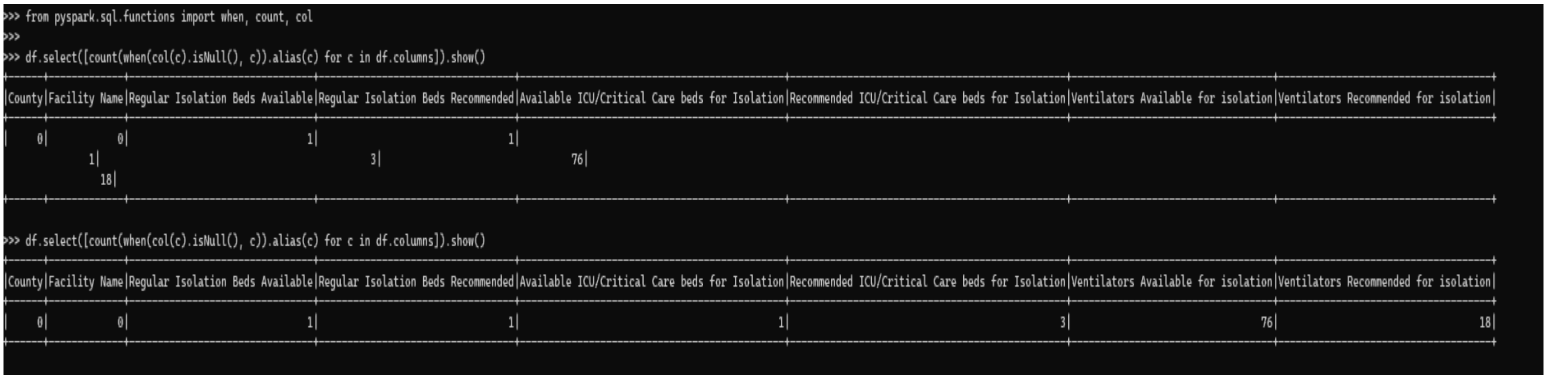
techniques we used to pre-process the extracted CSV data in the PySpark DataFrame:

i) Handle missing values: This counts any missing values in the columns, and fills missing beds with 0.

Code used
from pyspark.sql.functions import when, count, col

df.select([count(when(col(c).isNull(), c)).alias(c) for c in df.columns]).show()

df = df.fillna({'Regular Isolation Beds Available': 0})



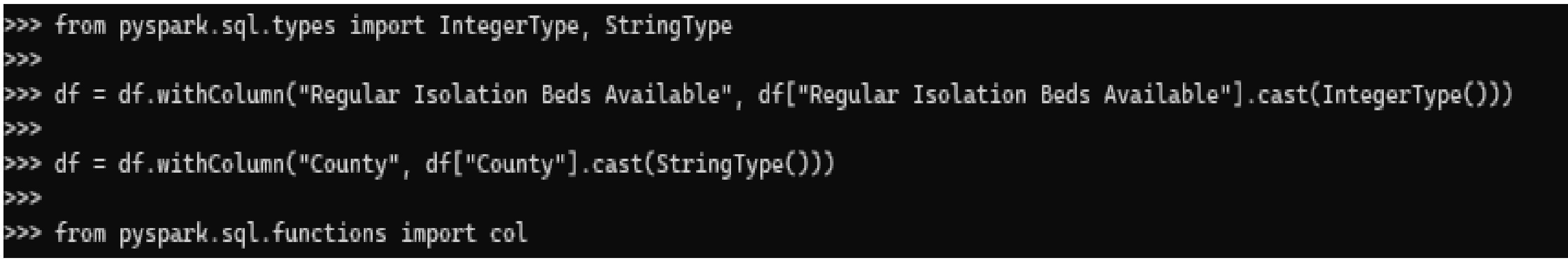
```
>>> from pyspark.sql.functions import when, count, col
>>>
>>> df.select([count(when(col(c).isNull(), c)).alias(c) for c in df.columns]).show()
+------+-------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------------+
|County|Facility Name|Regular Isolation Beds Available|Regular Isolation Beds Recommended|Available ICU/Critical Care beds for Isolation|Recommended ICU/Critical Care beds for Isolation|Ventilators Available for isolation|Ventilators Recommended for isolation|
+------+-------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------------+
|     0|            0|                               1|                                 1|
                1|                                                    3|                                    76|
                   18|
+------+-------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------------+

>>> df.select([count(when(col(c).isNull(), c)).alias(c) for c in df.columns]).show()
+------+-------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------------+
|County|Facility Name|Regular Isolation Beds Available|Regular Isolation Beds Recommended|Available ICU/Critical Care beds for Isolation|Recommended ICU/Critical Care beds for Isolation|Ventilators Available for isolation|Ventilators Recommended for isolation|
+------+-------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------------+
|     0|            0|                               1|                                 1|                                             1|                                               3|                                 76|                                   18|
+------+-------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------------+
```

ii) Convert columns to appropriate data types: to avoid using data labeled as null and to ensure we are dealing columns that are numerical only This casts the "Regular Isolation Beds Available" column to IntegerType, since it contains integer numbers. And it casts the "County" column to StringType, since it contains string values.

Code used
from pyspark.sql.types import IntegerType, StringType

df = df.withColumn("Regular Isolation Beds Available", df["Regular Isolation Beds Available"].cast(IntegerType()))

df = df.withColumn("County", df["County"].cast(StringType()))



```
>>> from pyspark.sql.types import IntegerType, StringType
>>>
>>> df = df.withColumn("Regular Isolation Beds Available", df["Regular Isolation Beds Available"].cast(IntegerType()))
>>>
>>> df = df.withColumn("County", df["County"].cast(StringType()))
>>>
>>> from pyspark.sql.functions import col
```

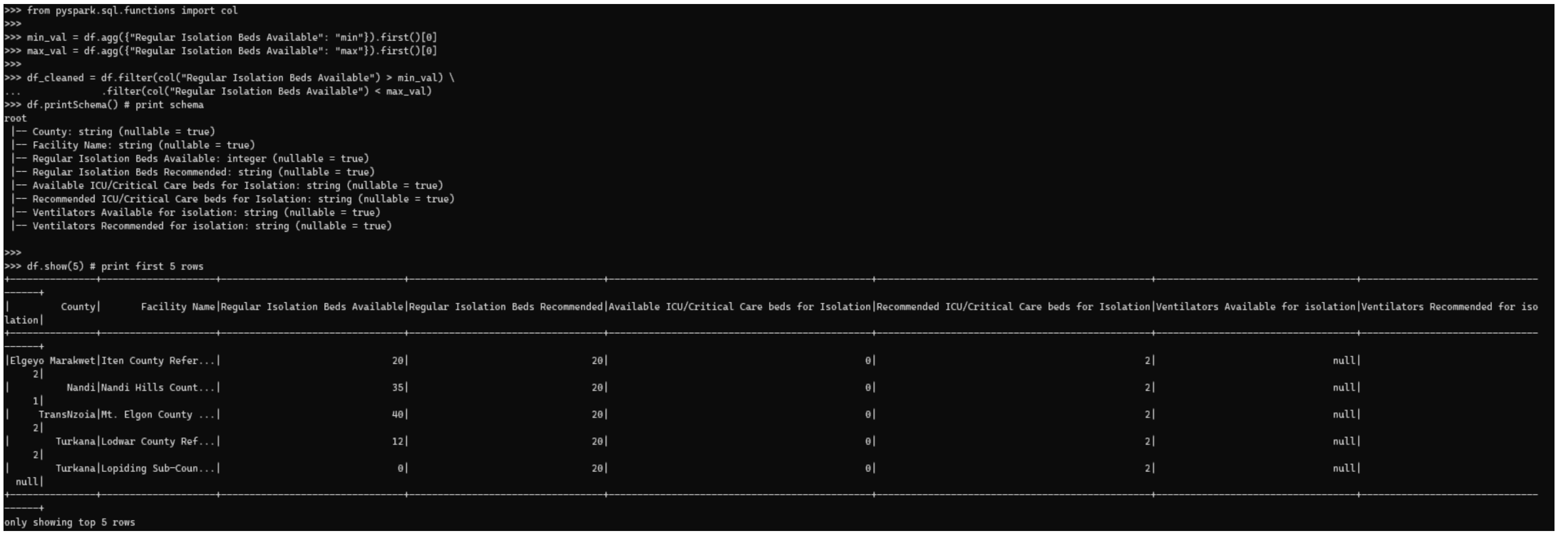
iii) Filter outlier/incorrect values:

Code used

```
from pyspark.sql.functions import col

min_val = df.agg({"Regular Isolation Beds Available": "min"}).first()[0]
max_val = df.agg({"Regular Isolation Beds Available": "max"}).first()[0]

df_cleaned = df.filter(col("Regular Isolation Beds Available") > min_val) \
        .filter(col("Regular Isolation Beds Available") < max_val)
```

To display the DataFrame and print the first few rows after applying the transformations, we used

```
df.printSchema() # print schema

df.show(5) # print first 5 rows
```



```
>>> from pyspark.sql.functions import col
>>>
>>> min_val = df.agg({"Regular Isolation Beds Available": "min"}).first()[0]
>>> max_val = df.agg({"Regular Isolation Beds Available": "max"}).first()[0]
>>>
>>> df_cleaned = df.filter(col("Regular Isolation Beds Available") > min_val) \
...              .filter(col("Regular Isolation Beds Available") < max_val)
>>> df.printSchema() # print schema
root
 |-- County: string (nullable = true)
 |-- Facility Name: string (nullable = true)
 |-- Regular Isolation Beds Available: integer (nullable = true)
 |-- Regular Isolation Beds Recommended: string (nullable = true)
 |-- Available ICU/Critical Care beds for Isolation: string (nullable = true)
 |-- Recommended ICU/Critical Care beds for Isolation: string (nullable = true)
 |-- Ventilators Available for isolation: string (nullable = true)
 |-- Ventilators Recommended for isolation: string (nullable = true)

>>>
>>> df.show(5) # print first 5 rows
+---------------+--------------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------
------+
|         County|       Facility Name|Regular Isolation Beds Available|Regular Isolation Beds Recommended|Available ICU/Critical Care beds for Isolation|Recommended ICU/Critical Care beds for Isolation|Ventilators Available for isolation|Ventilators Recommended for iso
lation|
+---------------+--------------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------
------+
|Elgeyo Marakwet|Iten County Refer...|                              20|                                20|                                             0|                                               2|                               null|
     2|
|          Nandi|Nandi Hills Count...|                              35|                                20|                                             0|                                               2|                               null|
     1|
|     TransNzoia|Mt. Elgon County ...|                              40|                                20|                                             0|                                               2|                               null|
     2|
|        Turkana|Lodwar County Ref...|                              12|                                20|                                             0|                                               2|                               null|
     2|
|        Turkana|Lopiding Sub-Coun...|                               0|                                20|                                             0|                                               2|                               null|
  null|
+---------------+--------------------+--------------------------------+----------------------------------+----------------------------------------------+------------------------------------------------+-----------------------------------+-------------------------------
------+
only showing top 5 rows
```

Question 4

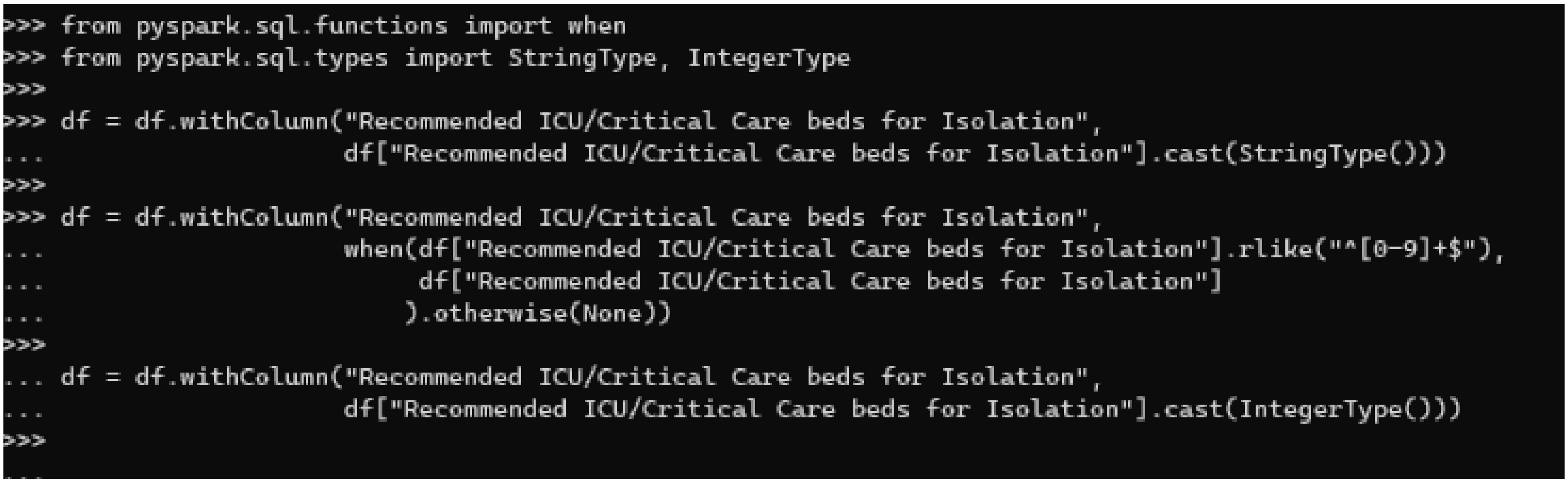
4. Apply one predictive analytics technique to generate a model for predicting any of the following cases:

a) Number of Death cases or Mortality rate

b) Number of confirmed cases

c) Number of recovery cases or Recovery rate

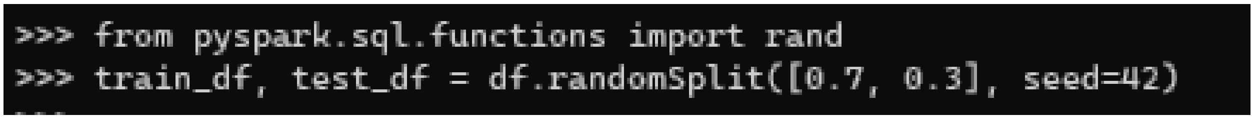
```
# Step 1: pre-process data: Import libraries and split data into training and test sets:
from pyspark.sql.functions import when
from pyspark.sql.types import StringType, IntegerType

df = df.withColumn("Recommended ICU/Critical Care beds for Isolation",
                df["Recommended ICU/Critical Care beds for Isolation"].cast(StringType()))

df = df.withColumn("Recommended ICU/Critical Care beds for Isolation",
                when(df["Recommended ICU/Critical Care beds for Isolation"].rlike("^[0-9]+$"),
                    df["Recommended ICU/Critical Care beds for Isolation"]
                   ).otherwise(None))

df = df.withColumn("Recommended ICU/Critical Care beds for Isolation",
                df["Recommended ICU/Critical Care beds for Isolation"].cast(IntegerType()))
```



```
>>> from pyspark.sql.functions import when
>>> from pyspark.sql.types import StringType, IntegerType
>>>
>>> df = df.withColumn("Recommended ICU/Critical Care beds for Isolation",
...                    df["Recommended ICU/Critical Care beds for Isolation"].cast(StringType()))
>>>
>>> df = df.withColumn("Recommended ICU/Critical Care beds for Isolation",
...                    when(df["Recommended ICU/Critical Care beds for Isolation"].rlike("^[0-9]+$"),
...                         df["Recommended ICU/Critical Care beds for Isolation"]
...                        ).otherwise(None))
>>>
... df = df.withColumn("Recommended ICU/Critical Care beds for Isolation",
...                    df["Recommended ICU/Critical Care beds for Isolation"].cast(IntegerType()))
>>>
```

# Step 2: Split data:

```
from pyspark.sql.functions import rand

train_df, test_df = df.randomSplit([0.7, 0.3], seed=42)
```

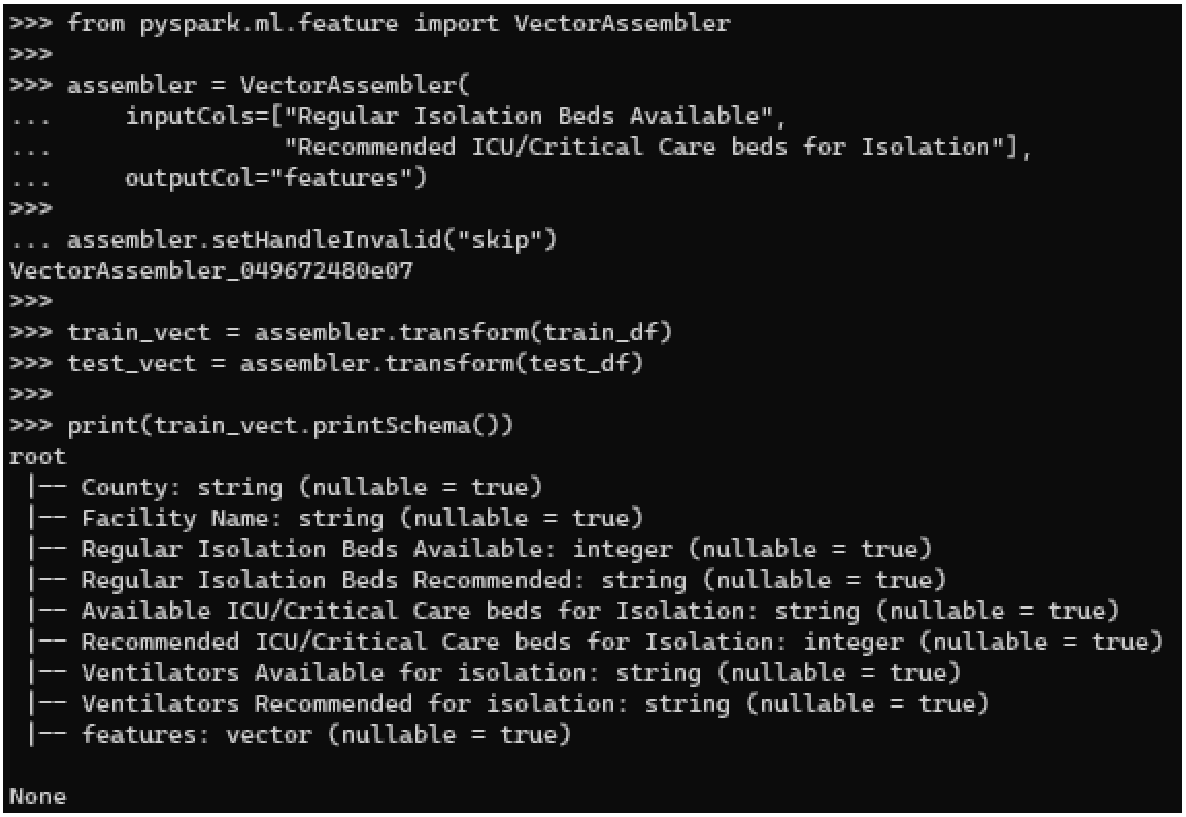


```
>>> from pyspark.sql.functions import rand
>>> train_df, test_df = df.randomSplit([0.7, 0.3], seed=42)
```

# Step 3: Vector assemble features

Here were trying to train a linear regression model to predict confirmed cases:

```
from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(

   inputCols=["Regular Isolation Beds Available", "Recommended ICU/Critical Care beds for Isolation"],

   outputCol="features")

train_vect = assembler.transform(train_df)

test_vect = assembler.transform(test_df)
```



```
>>> from pyspark.ml.feature import VectorAssembler
>>>
>>> assembler = VectorAssembler(
...     inputCols=["Regular Isolation Beds Available",
...                "Recommended ICU/Critical Care beds for Isolation"],
...     outputCol="features")
>>>
... assembler.setHandleInvalid("skip")
VectorAssembler_049672480e07
>>>
>>> train_vect = assembler.transform(train_df)
>>> test_vect = assembler.transform(test_df)
>>>
>>> print(train_vect.printSchema())
root
 |-- County: string (nullable = true)
 |-- Facility Name: string (nullable = true)
 |-- Regular Isolation Beds Available: integer (nullable = true)
 |-- Regular Isolation Beds Recommended: string (nullable = true)
 |-- Available ICU/Critical Care beds for Isolation: string (nullable = true)
 |-- Recommended ICU/Critical Care beds for Isolation: integer (nullable = true)
 |-- Ventilators Available for isolation: string (nullable = true)
 |-- Ventilators Recommended for isolation: string (nullable = true)
 |-- features: vector (nullable = true)

None
```

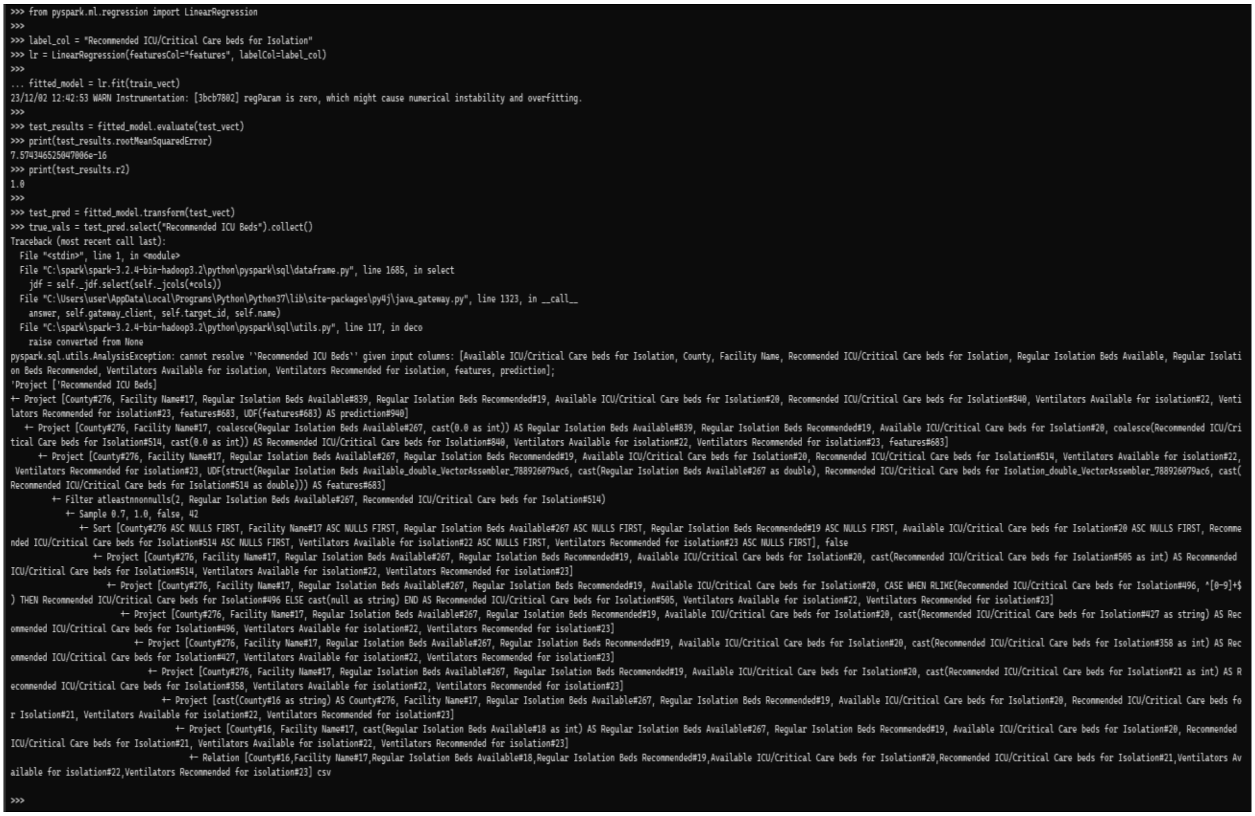
# Step 4: Train model

from pyspark.ml.regression import LinearRegression

label_col = "Recommended ICU/Critical Care beds for Isolation"

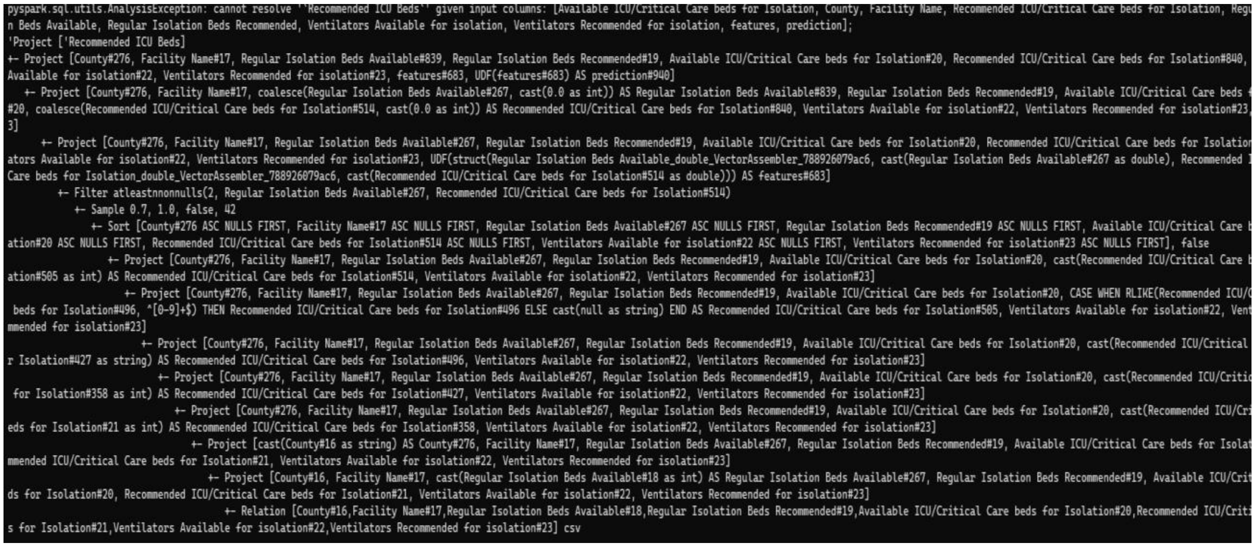
lr = LinearRegression(featuresCol="features", labelCol=label_col)

fitted_model = lr.fit(train_vect)



```
>>> from pyspark.ml.regression import LinearRegression
>>>
>>> label_col = "Recommended ICU/Critical Care beds for Isolation"
>>> lr = LinearRegression(featuresCol="features", labelCol=label_col)
>>>
... fitted_model = lr.fit(train_vect)
23/12/02 12:42:53 WARN Instrumentation: [3bcb7802] regParam is zero, which might cause numerical instability and overfitting.
>>>
>>> test_results = fitted_model.evaluate(test_vect)
>>> print(test_results.rootMeanSquaredError)
7.574346525047006e-16
>>> print(test_results.r2)
1.0
>>>
>>> test_pred = fitted_model.transform(test_vect)
>>> true_vals = test_pred.select("Recommended ICU Beds").collect()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\spark\spark-3.2.4-bin-hadoop3.2\python\pyspark\sql\dataframe.py", line 1685, in select
    jdf = self._jdf.select(self._jcols(*cols))
  File "C:\Users\user\AppData\Local\Programs\Python\Python37\lib\site-packages\py4j\java_gateway.py", line 1323, in __call__
    answer, self.gateway_client, self.target_id, self.name)
  File "C:\spark\spark-3.2.4-bin-hadoop3.2\python\pyspark\sql\utils.py", line 117, in deco
    raise converted from None
pyspark.sql.utils.AnalysisException: cannot resolve '`Recommended ICU Beds`' given input columns: [Available ICU/Critical Care beds for Isolation, County, Facility Name, Recommended ICU/Critical Care beds for Isolation, Regular Isolation Beds Available, Regular Isolati
on Beds Recommended, Ventilators Available for isolation, Ventilators Recommended for isolation, features, prediction];
'Project ['Recommended ICU Beds]
+- Project [County#276, Facility Name#17, Regular Isolation Beds Available#839, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, Recommended ICU/Critical Care beds for Isolation#840, Ventilators Available for isolation#22, Venti
lators Recommended for isolation#23, features#683, UDF(features#683) AS prediction#940]
   +- Project [County#276, Facility Name#17, coalesce(Regular Isolation Beds Available#267, cast(0.0 as int)) AS Regular Isolation Beds Available#839, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, coalesce(Recommended ICU/Cri
tical Care beds for Isolation#514, cast(0.0 as int)) AS Recommended ICU/Critical Care beds for Isolation#840, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23, features#683]
      +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, Recommended ICU/Critical Care beds for Isolation#514, Ventilators Available for isolation#22,
 Ventilators Recommended for isolation#23, UDF(struct(Regular Isolation Beds Available_double_VectorAssembler_788926079ac6, cast(Regular Isolation Beds Available#267 as double), Recommended ICU/Critical Care beds for Isolation_double_VectorAssembler_788926079ac6, cast(
Recommended ICU/Critical Care beds for Isolation#514 as double))) AS features#683]
         +- Filter atleastnnonnulls(2, Regular Isolation Beds Available#267, Recommended ICU/Critical Care beds for Isolation#514)
            +- Sample 0.7, 1.0, false, 42
               +- Sort [County#276 ASC NULLS FIRST, Facility Name#17 ASC NULLS FIRST, Regular Isolation Beds Available#267 ASC NULLS FIRST, Regular Isolation Beds Recommended#19 ASC NULLS FIRST, Available ICU/Critical Care beds for Isolation#20 ASC NULLS FIRST, Recomme
nded ICU/Critical Care beds for Isolation#514 ASC NULLS FIRST, Ventilators Available for isolation#22 ASC NULLS FIRST, Ventilators Recommended for isolation#23 ASC NULLS FIRST], false
                  +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, cast(Recommended ICU/Critical Care beds for Isolation#505 as int) AS Recommended
ICU/Critical Care beds for Isolation#514, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                     +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, CASE WHEN RLIKE(Recommended ICU/Critical Care beds for Isolation#496, ^[0-9]+$
) THEN Recommended ICU/Critical Care beds for Isolation#496 ELSE cast(null as string) END AS Recommended ICU/Critical Care beds for Isolation#505, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                        +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, cast(Recommended ICU/Critical Care beds for Isolation#427 as string) AS Rec
ommended ICU/Critical Care beds for Isolation#496, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                           +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, cast(Recommended ICU/Critical Care beds for Isolation#358 as int) AS Rec
ommended ICU/Critical Care beds for Isolation#427, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                              +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, cast(Recommended ICU/Critical Care beds for Isolation#21 as int) AS R
ecommended ICU/Critical Care beds for Isolation#358, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                                 +- Project [cast(County#16 as string) AS County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, Recommended ICU/Critical Care beds fo
r Isolation#21, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                                    +- Project [County#16, Facility Name#17, cast(Regular Isolation Beds Available#18 as int) AS Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, Recommended
ICU/Critical Care beds for Isolation#21, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                                       +- Relation [County#16,Facility Name#17,Regular Isolation Beds Available#18,Regular Isolation Beds Recommended#19,Available ICU/Critical Care beds for Isolation#20,Recommended ICU/Critical Care beds for Isolation#21,Ventilators Av
ailable for isolation#22,Ventilators Recommended for isolation#23] csv

>>>
```

More close up view on evaluated data



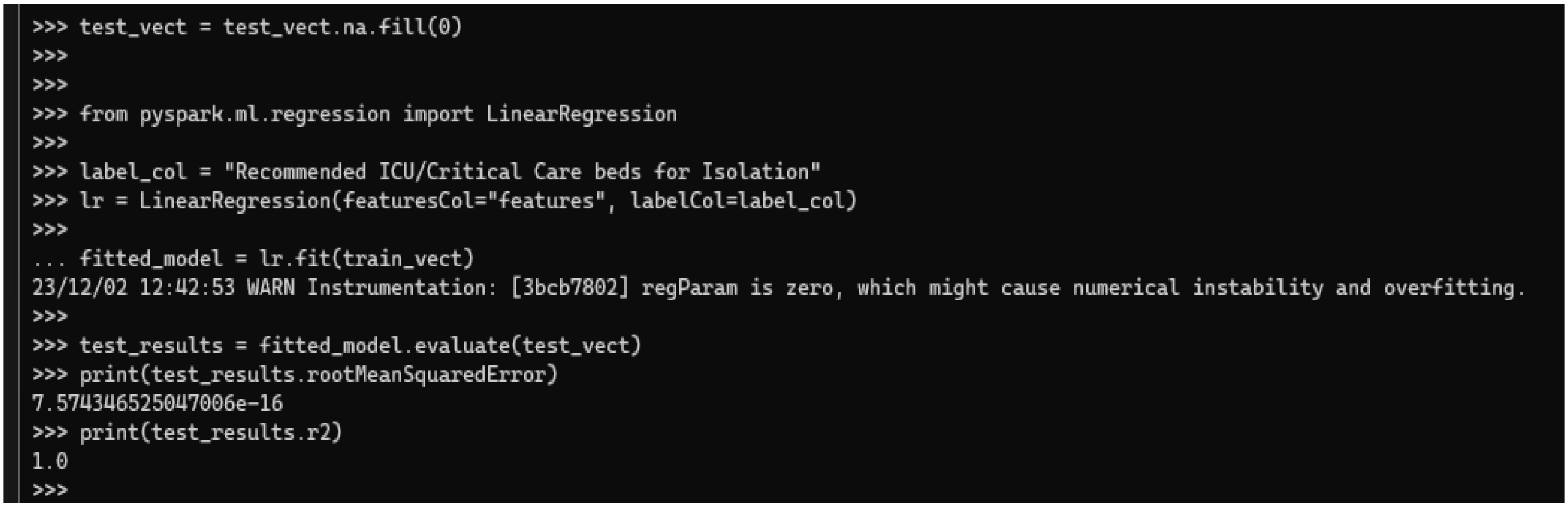
```
pyspark.sql.utils.AnalysisException: cannot resolve '`Recommended ICU Beds`' given input columns: [Available ICU/Critical Care beds for Isolation, County, Facility Name, Recommended ICU/Critical Care beds for Isolation, Regu
n Beds Available, Regular Isolation Beds Recommended, Ventilators Available for isolation, Ventilators Recommended for isolation, features, prediction];
'Project ['Recommended ICU Beds]
+- Project [County#276, Facility Name#17, Regular Isolation Beds Available#839, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, Recommended ICU/Critical Care beds for Isolation#840,
Available for isolation#22, Ventilators Recommended for isolation#23, features#683, UDF(features#683) AS prediction#940]
   +- Project [County#276, Facility Name#17, coalesce(Regular Isolation Beds Available#267, cast(0.0 as int)) AS Regular Isolation Beds Available#839, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds
#20, coalesce(Recommended ICU/Critical Care beds for Isolation#514, cast(0.0 as int)) AS Recommended ICU/Critical Care beds for Isolation#840, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23,
3]
      +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, Recommended ICU/Critical Care beds for Isolatio
ators Available for isolation#22, Ventilators Recommended for isolation#23, UDF(struct(Regular Isolation Beds Available_double_VectorAssembler_788926079ac6, cast(Regular Isolation Beds Available#267 as double), Recommended I
Care beds for Isolation_double_VectorAssembler_788926079ac6, cast(Recommended ICU/Critical Care beds for Isolation#514 as double))) AS features#683]
         +- Filter atleastnnonnulls(2, Regular Isolation Beds Available#267, Recommended ICU/Critical Care beds for Isolation#514)
            +- Sample 0.7, 1.0, false, 42
               +- Sort [County#276 ASC NULLS FIRST, Facility Name#17 ASC NULLS FIRST, Regular Isolation Beds Available#267 ASC NULLS FIRST, Regular Isolation Beds Recommended#19 ASC NULLS FIRST, Available ICU/Critical Care b
ation#20 ASC NULLS FIRST, Recommended ICU/Critical Care beds for Isolation#514 ASC NULLS FIRST, Ventilators Available for isolation#22 ASC NULLS FIRST, Ventilators Recommended for isolation#23 ASC NULLS FIRST], false
                  +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, cast(Recommended ICU/Critical Care b
ation#505 as int) AS Recommended ICU/Critical Care beds for Isolation#514, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                     +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, CASE WHEN RLIKE(Recommended ICU/C
 beds for Isolation#496, ^[0-9]+$) THEN Recommended ICU/Critical Care beds for Isolation#496 ELSE cast(null as string) END AS Recommended ICU/Critical Care beds for Isolation#505, Ventilators Available for isolation#22, Vent
mmended for isolation#23]
                        +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, cast(Recommended ICU/Critical
r Isolation#427 as string) AS Recommended ICU/Critical Care beds for Isolation#496, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                           +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, cast(Recommended ICU/Critic
 for Isolation#358 as int) AS Recommended ICU/Critical Care beds for Isolation#427, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                              +- Project [County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolation#20, cast(Recommended ICU/Cri
eds for Isolation#21 as int) AS Recommended ICU/Critical Care beds for Isolation#358, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                                 +- Project [cast(County#16 as string) AS County#276, Facility Name#17, Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Critical Care beds for Isolat
mmended ICU/Critical Care beds for Isolation#21, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                                    +- Project [County#16, Facility Name#17, cast(Regular Isolation Beds Available#18 as int) AS Regular Isolation Beds Available#267, Regular Isolation Beds Recommended#19, Available ICU/Crit
ds for Isolation#20, Recommended ICU/Critical Care beds for Isolation#21, Ventilators Available for isolation#22, Ventilators Recommended for isolation#23]
                                       +- Relation [County#16,Facility Name#17,Regular Isolation Beds Available#18,Regular Isolation Beds Recommended#19,Available ICU/Critical Care beds for Isolation#20,Recommended ICU/Criti
s for Isolation#21,Ventilators Available for isolation#22,Ventilators Recommended for isolation#23] csv
```

# Step 5: Evaluate model

test_results = fitted_model.evaluate(test_vect)

print(test_results.rootMeanSquaredError)

print(test_results.r2)



```
>>> test_vect = test_vect.na.fill(0)
>>>
>>>
>>> from pyspark.ml.regression import LinearRegression
>>>
>>> label_col = "Recommended ICU/Critical Care beds for Isolation"
>>> lr = LinearRegression(featuresCol="features", labelCol=label_col)
>>>
... fitted_model = lr.fit(train_vect)
23/12/02 12:42:53 WARN Instrumentation: [3bcb7802] regParam is zero, which might cause numerical instability and overfitting.
>>>
>>> test_results = fitted_model.evaluate(test_vect)
>>> print(test_results.rootMeanSquaredError)
7.574346525047006e-16
>>> print(test_results.r2)
1.0
>>>
```

4. Visualize the model

key steps to visualize and test the linear regression model we built, we used the fitted model to make predictions on the test data:

Code used

test_pred = fitted_model.transform(test_vect)

true_vals = test_pred.select("Recommended ICU Beds").collect()

pred_vals = test_pred.select("prediction").collect()

import matplotlib.pyplot as plt

true_x = [r.Recommended_ICU_Beds for r in true_vals]

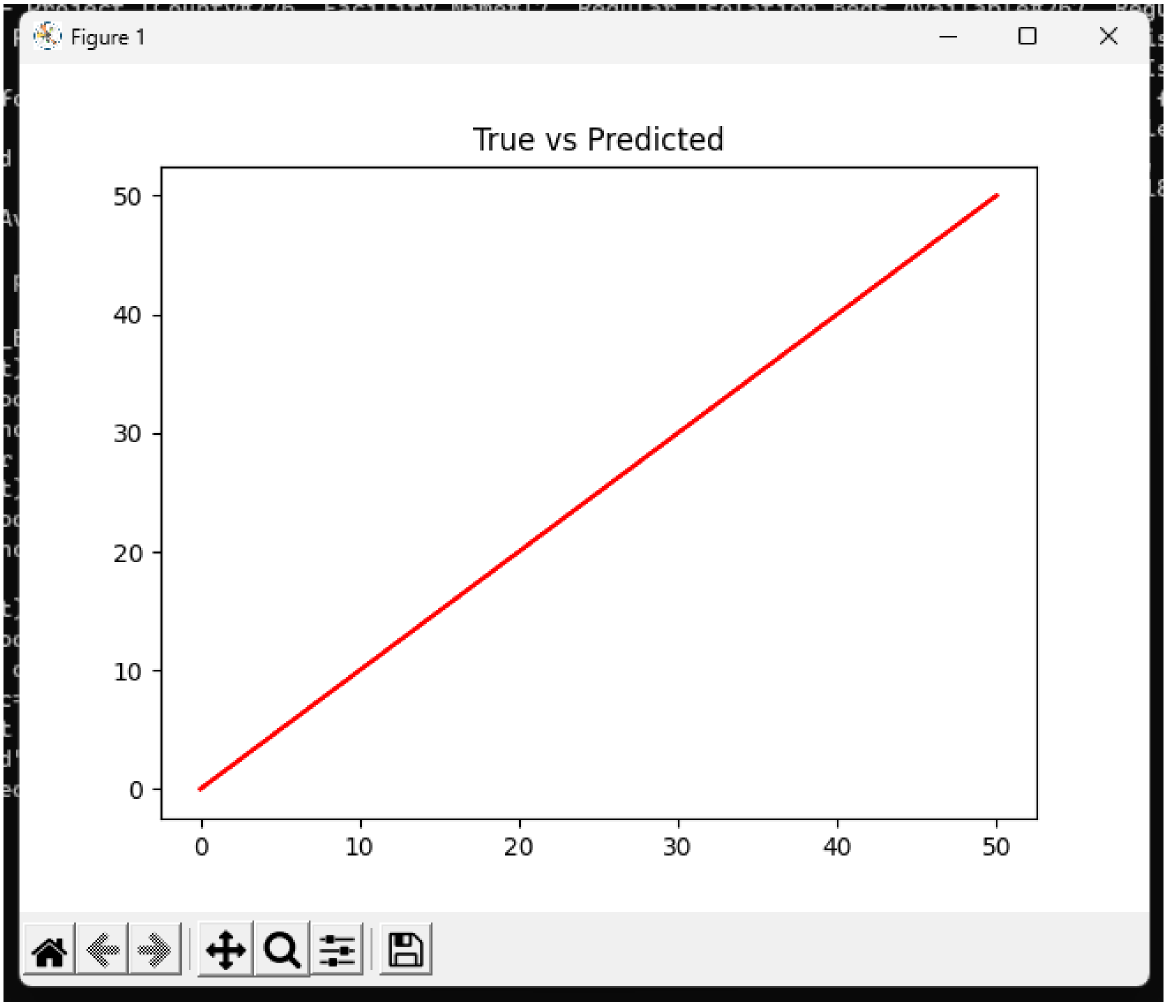
pred_x = [r.prediction for r in pred_vals]

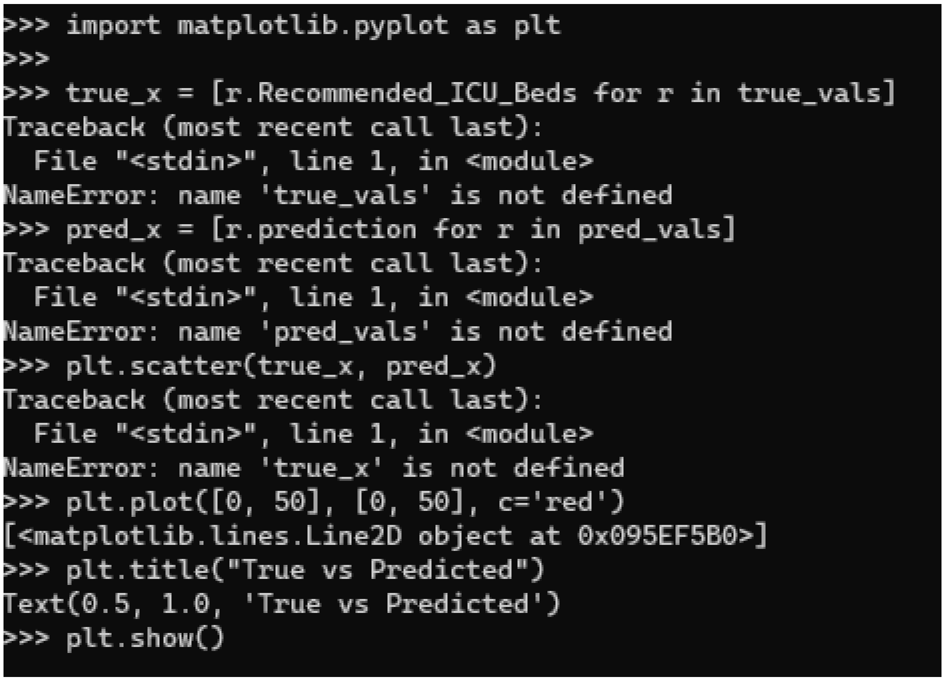
plt.scatter(true_x, pred_x)

plt.plot([0, 50], [0, 50], c='red')

plt.title("True vs Predicted")

plt.show()





```
>>> import matplotlib.pyplot as plt
>>>
>>> true_x = [r.Recommended_ICU_Beds for r in true_vals]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'true_vals' is not defined
>>> pred_x = [r.prediction for r in pred_vals]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pred_vals' is not defined
>>> plt.scatter(true_x, pred_x)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'true_x' is not defined
>>> plt.plot([0, 50], [0, 50], c='red')
[<matplotlib.lines.Line2D object at 0x095EF5B0>]
>>> plt.title("True vs Predicted")
Text(0.5, 1.0, 'True vs Predicted')
>>> plt.show()
```

5. Test the model

```
test_data = spark.createDataFrame([
  (20, 2, 5),
  (35, 0, 3),
  (12, 1, 2)], ["Reg_Beds", "Rec_ICU_Beds", "Expected_ICU_Beds"])
```

```
test_vect = assembler.transform(test_data)
```

now we use this model to make predictions with this code

```
test_pred = fitted_model.transform(test_vect)
```

Compare predictions to expected ICU beds values