Thasina Tabashum

1. Importing Data

In [1]:
```
from pyspark.sql import SparkSession
```

In [2]:
```
spark = SparkSession.builder.appName('cruise').getOrCreate()
```

In [3]:
```
df = spark.read.csv('gs://spark2_thasina/cruise_ship_info.csv',inferSchema=True,h
```

In [4]:
```
df.printSchema()
```

```
root
 |-- Ship_name: string (nullable = true)
 |-- Cruise_line: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Tonnage: double (nullable = true)
 |-- passengers: double (nullable = true)
 |-- length: double (nullable = true)
 |-- cabins: double (nullable = true)
 |-- passenger_density: double (nullable = true)
 |-- crew: double (nullable = true)
```

In [6]:
```
df.show()
```

```
+----------+----------+---+-----------------+----------+------+------+------
----------+----+
|  Ship_name|Cruise_line|Age|          Tonnage|passengers|length|cabins|passen
ger_density|crew|
+----------+----------+---+-----------------+----------+------+------+------
----------+----+
|    Journey|    Azamara|  6|30.276999999999997|      6.94|  5.94|  3.55|
42.64|3.55|
|      Quest|    Azamara|  6|30.276999999999997|      6.94|  5.94|  3.55|
42.64|3.55|
|Celebration|   Carnival| 26|           47.262|     14.86|  7.22|  7.43|
31.8| 6.7|
|   Conquest|   Carnival| 11|            110.0|     29.74|  9.53| 14.88|
36.99|19.1|
|    Destiny|   Carnival| 17|          101.353|     26.42|  8.92| 13.21|
38.36|10.0|
|    Ecstasy|   Carnival| 22|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|
|    Elation|   Carnival| 15|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|
|    Fantasy|   Carnival| 23|           70.367|     20.56|  8.55| 10.22|
34.23| 9.2|
|Fascination|   Carnival| 19|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|
|    Freedom|   Carnival|  6|110.23899999999999|      37.0|  9.51| 14.87|
29.79|11.5|
|      Glory|   Carnival| 10|            110.0|     29.74|  9.51| 14.87|
36.99|11.6|
|    Holiday|   Carnival| 28|           46.052|     14.52|  7.27|  7.26|
31.72| 6.6|
|Imagination|   Carnival| 18|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|
|Inspiration|   Carnival| 17|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|
|     Legend|   Carnival| 11|             86.0|     21.24|  9.63| 10.62|
40.49| 9.3|
|   Liberty*|   Carnival|  8|            110.0|     29.74|  9.51| 14.87|
36.99|11.6|
|    Miracle|   Carnival|  9|             88.5|     21.24|  9.63| 10.62|
41.67|10.3|
|   Paradise|   Carnival| 15|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|
|      Pride|   Carnival| 12|             88.5|     21.24|  9.63| 11.62|
41.67| 9.3|
|  Sensation|   Carnival| 20|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|
+----------+----------+---+-----------------+----------+------+------+------
----------+----+
only showing top 20 rows
```

In [7]:
```
df.describe().show()
```

```
+-------+---------+-----------+------------------+------------------+----------
-------+-----------------+-----------------+----------------+---------------
-+
|summary|Ship_name|Cruise_line|               Age|           Tonnage|       pas
sengers|           length|            cabins|passenger_density|              cre
w|
+-------+---------+-----------+------------------+------------------+----------
-------+-----------------+-----------------+----------------+---------------
-+
|  count|      158|        158|               158|               158|
158|              158|              158|             158|              158|
|   mean| Infinity|       null|15.689873417721518| 71.28467088607599|18.4574050
6329114|8.130632911392404| 8.830000000000005|39.90094936708861|7.79417721518987
3|
| stddev|      NaN|       null| 7.615691058751413|37.229540025907866|9.67709477
5143416|1.793473548054825|4.4714172221480615| 8.63921711391542|3.50348656462703
4|
|    min|Adventure|    Azamara|                 4|             2.329|
0.66|             2.79|             0.33|             17.7|             0.59|
|    max|Zuiderdam|   Windstar|                48|             220.0|
54.0|            11.82|             27.0|            71.43|             21.0|
+-------+---------+-----------+------------------+------------------+----------
-------+-----------------+-----------------+----------------+---------------
-+
```

2. Data Preprocessing

In [8]:
```
from pyspark.ml.feature import StringIndexer
```

In [9]:
```
indexer = StringIndexer(inputCol="Cruise_line", outputCol="Cruise_line_out")
indexed = indexer.fit(df).transform(df)
indexed.show()
```

```
+----------+----------+---+-----------------+----------+------+------+------
----------+----+--------------+
|  Ship_name|Cruise_line|Age|          Tonnage|passengers|length|cabins|passen
ger_density|crew|Cruise_line_out|
+----------+----------+---+-----------------+----------+------+------+------
----------+----+--------------+
|    Journey|    Azamara|  6|30.276999999999997|      6.94|  5.94|  3.55|
42.64|3.55|          16.0|
|      Quest|    Azamara|  6|30.276999999999997|      6.94|  5.94|  3.55|
42.64|3.55|          16.0|
|Celebration|   Carnival| 26|           47.262|     14.86|  7.22|  7.43|
31.8| 6.7|           1.0|
|   Conquest|   Carnival| 11|            110.0|     29.74|  9.53| 14.88|
36.99|19.1|           1.0|
|    Destiny|   Carnival| 17|          101.353|     26.42|  8.92| 13.21|
38.36|10.0|           1.0|
|    Ecstasy|   Carnival| 22|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|           1.0|
|    Elation|   Carnival| 15|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|           1.0|
|    Fantasy|   Carnival| 23|           70.367|     20.56|  8.55| 10.22|
34.23| 9.2|           1.0|
|Fascination|   Carnival| 19|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|           1.0|
|    Freedom|   Carnival|  6|110.23899999999999|      37.0|  9.51| 14.87|
29.79|11.5|           1.0|
|      Glory|   Carnival| 10|            110.0|     29.74|  9.51| 14.87|
36.99|11.6|           1.0|
|    Holiday|   Carnival| 28|           46.052|     14.52|  7.27|  7.26|
31.72| 6.6|           1.0|
|Imagination|   Carnival| 18|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|           1.0|
|Inspiration|   Carnival| 17|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|           1.0|
|     Legend|   Carnival| 11|             86.0|     21.24|  9.63| 10.62|
40.49| 9.3|           1.0|
|   Liberty*|   Carnival|  8|            110.0|     29.74|  9.51| 14.87|
36.99|11.6|           1.0|
|    Miracle|   Carnival|  9|             88.5|     21.24|  9.63| 10.62|
41.67|10.3|           1.0|
|   Paradise|   Carnival| 15|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|           1.0|
|      Pride|   Carnival| 12|             88.5|     21.24|  9.63| 11.62|
41.67| 9.3|           1.0|
|  Sensation|   Carnival| 20|           70.367|     20.52|  8.55|  10.2|
34.29| 9.2|           1.0|
+----------+----------+---+-----------------+----------+------+------+------
----------+----+--------------+
only showing top 20 rows
```

In [10]:
```
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
```

In [11]:
```
assembler = VectorAssembler(
    inputCols=["Age", "Tonnage", "passengers","length","cabins","passenger_densit
    outputCol="features")
```

In [14]:
```
output = assembler.transform(indexed)
```

In [15]:
```
output
```

Out[15]:
```
DataFrame[Ship_name: string, Cruise_line: string, Age: int, Tonnage: double, pa
ssengers: double, length: double, cabins: double, passenger_density: double, cr
ew: double, Cruise_line_out: double, features: vector]
```

In [25]:
```
X = output.select("features", "crew")
```

In [17]:
```
from pyspark.ml.regression import LinearRegression
```

In [26]:
```
train_data,test_data = X.randomSplit([0.8,0.2])
```

In [27]:
```
train_data.show()
```

```
+--------------------+----+
|            features|crew|
+--------------------+----+
|[4.0,220.0,54.0,1...|21.0|
|[5.0,86.0,21.04,9...| 8.0|
|[5.0,115.0,35.74,...|12.2|
|[5.0,122.0,28.5,1...| 6.7|
|[5.0,160.0,36.34,...|13.6|
|[6.0,30.276999999...|3.55|
|[6.0,30.276999999...|3.55|
|[6.0,90.0,20.0,9....| 9.0|
|[6.0,112.0,38.0,9...|10.9|
|[6.0,113.0,37.82,...|12.0|
|[7.0,89.6,25.5,9....|9.87|
|[7.0,116.0,31.0,9...|12.0|
|[7.0,158.0,43.7,1...|13.6|
|[8.0,77.499,19.5,...| 9.0|
|[8.0,91.0,22.44,9...|11.0|
|[8.0,110.0,29.74,...|11.6|
|[9.0,59.058,17.0,...| 7.4|
|[9.0,81.0,21.44,9...|10.0|
|[9.0,85.0,19.68,9...|8.69|
|[9.0,88.5,21.24,9...|10.3|
+--------------------+----+
only showing top 20 rows
```

3. Model and Training

In [32]:
```
lr = LinearRegression(featuresCol='features', labelCol='crew', predictionCol='pre
```

In [33]:
```
lrModel = lr.fit(train_data)
```

In [34]:
```
# Print the coefficients and intercept for linear regression
print("Coefficients: {}".format(str(lrModel.coefficients))) # For each feature...
print('\n')
print("Intercept:{}".format(str(lrModel.intercept)))
```

```
Coefficients: [-0.010428549768007195,0.005292036735701787,-0.11703667306313584,
0.44934062622045673,0.8080309363129383,0.0017881549812901224,0.0500188903893847
15]


Intercept:-1.3627071992
```

In [35]:
```
trainingSummary = lrModel.summary
```

In [36]:
```
trainingSummary.residuals.show()
print("RMSE: {}".format(trainingSummary.rootMeanSquaredError))
print("r2: {}".format(trainingSummary.r2))
```

```
+-------------------+
|          residuals|
+-------------------+
|0.35926274551887616|
|-1.2648566057723283|
|0.25834399761489735|
|0.24997558722187563|
|-1.3360855707568682|
|-0.7868563056686497|
|-0.7868563056686497|
|-1.1873036485617678|
|-0.5165208194827571|
|0.24586643655762153|
|-1.2175501355648493|
| -0.571253922730989|
|-0.2918164451288181|
|0.42246671786635304|
| 0.9160338034293307|
|-0.4601299081579242|
|-0.1753341898446772|
|0.46902458048286455|
|-0.6797369451622455|
| 0.7411077492621114|
+-------------------+
only showing top 20 rows

RMSE: 1.01514219425
r2: 0.920457648563
```

4.Testing

In [37]:
```
test_results = lrModel.evaluate(test_data)
```

In [38]:
```
test_results.residuals.show()
print("RMSE: {}".format(test_results.rootMeanSquaredError))
```

```
+-------------------+
|          residuals|
+-------------------+
|-0.07734336032899591|
|  0.5471643012501062|
|  0.2803091578298833|
| -0.3606592763054852|
| -0.5108505068047631|
|   1.115916449542194|
| -0.7813111648706208|
| -0.4359632396349733|
|   0.810875254744909|
|-0.30189733080879577|
|-0.43617648761404126|
|  0.7902626054059141|
| 0.23056196861298162|
| -0.6031523540594375|
|  0.5906087925906292|
| -0.6353785989352421|
| -0.4452875542906707|
|  0.6107042625505059|
| -0.5546133212229263|
|-0.40705406138465516|
+-------------------+
only showing top 20 rows

RMSE: 0.580394806408
```

In [39]:
```
print("RMSE: {}".format(test_results.rootMeanSquaredError))
print("MSE: {}".format(test_results.meanSquaredError))
print("R2: {}".format(test_results.r2))
```

```
RMSE: 0.580394806408
MSE: 0.336858131305
R2: 0.962617522777
```

In [40]:
```
from pyspark.sql.functions import corr
```

In [41]:
```
df.select(corr('crew','passengers')).show()
```

```
+---------------------+
|corr(crew, passengers)|
+---------------------+
|    0.9152341306065384|
+---------------------+
```

In [43]:
```
df.select(corr('crew','passenger_density')).show()
```

```
+----------------------------+
|corr(crew, passenger_density)|
+----------------------------+
|         -0.15550928421699717|
+----------------------------+
```