Linear Regression Project - SOLUTIONS

Congratulations! You've been contracted by Hyundai Heavy Industries to help them build a predictive model for some ships. Hyundai Heavy Industries (http://www.hyundai.eu/en) is one of the world's largest ship manufacturing companies and builds cruise liners.

You've been flown to their headquarters in Ulsan, South Korea to help them give accurate estimates of how many crew members a ship will require.

They are currently building new ships for some customers and want you to create a model and use it to predict how many crew members the ships will need.

Here is what the data looks like so far:

```
Description: Measurements of ship size, capacity, crew, and age for 158 cruise
ships.


Variables/Columns
Ship Name     1-20
Cruise Line   21-40
Age (as of 2013)   46-48
Tonnage (1000s of tons)   50-56
passengers (100s)   58-64
Length (100s of feet)  66-72
Cabins  (100s)   74-80
Passenger Density   82-88
Crew  (100s)   90-96
```

It is saved in a csv file for you called "cruise_ship_info.csv". Your job is to create a regression model that will help predict how many crew members will be needed for future ships. The client also mentioned that they have found that particular cruise lines will differ in acceptable crew counts, so it is most likely an important feature to include in your analysis!

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

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

In [3]:
```
df = spark.read.csv('cruise_ship_info.csv',inferSchema=True,header=True)
```

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 [5]:
```
df.show()
```

```
+-----------+-----------+---+-----------------+----------+------+------+-----------------+----+
|  Ship_name|Cruise_line|Age|          Tonnage|passengers|length|cabins|passenger_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 [6]:
```
df.describe().show()
```

```
+-------+------------------+------------------+-----------------+-----------------+------------------+-----------------+----
-------------+-----------------+
|summary|               Age|          Tonnage|       passengers|           length|            cabins|pass
enger_density|            crew|
+-------+------------------+------------------+-----------------+-----------------+------------------+----
-------------+-----------------+
|  count|               158|               158|              158|              158|               158|
          158|             158|
|   mean|15.689873417721518| 71.28467088607599|18.45740506329114|8.130632911392404| 8.830000000000005|39.9
0094936708861|7.794177215189873|
| stddev| 7.615691058751413|37.229540025907866|9.677094775143416|1.793473548054825|4.4714172221480615| 8.6
3921711391542|3.503486564627034|
|    min|                 4|             2.329|             0.66|             2.79|              0.33|
          17.7|             0.59|
|    max|                48|             220.0|             54.0|            11.82|              27.0|
          71.43|             21.0|
+-------+------------------+------------------+-----------------+-----------------+------------------+----
-------------+-----------------+
```

Dealing with the Cruise_line categorical variable

Ship Name is a useless arbitrary string, but the cruise_line itself may be useful. Let's make it into a categorical variable!

In [7]:
```
df.groupBy('Cruise_line').count().show()
```

```
+-----------------+-----+
|      Cruise_line|count|
+-----------------+-----+
|            Costa|   11|
|              P&O|    6|
|           Cunard|    3|
|Regent_Seven_Seas|    5|
|              MSC|    8|
|         Carnival|   22|
|          Crystal|    2|
|           Orient|    1|
|         Princess|   17|
|        Silversea|    4|
|          Seabourn|    3|
| Holland_American|   14|
|         Windstar|    3|
|           Disney|    2|
|        Norwegian|   13|
|          Oceania|    3|
|          Azamara|    2|
|        Celebrity|   10|
|             Star|    6|
|  Royal_Caribbean|   23|
+-----------------+-----+
```

In [8]:
```
from pyspark.ml.feature import StringIndexer
indexer = StringIndexer(inputCol="Cruise_line", outputCol="cruise_cat")
indexed = indexer.fit(df).transform(df)
indexed.head(5)
```

Out[8]:
```
[Row(Ship_name='Journey', Cruise_line='Azamara', Age=6, Tonnage=30.276999999999997, passengers=6.94, lengt
h=5.94, cabins=3.55, passenger_density=42.64, crew=3.55, cruise_cat=16.0),
 Row(Ship_name='Quest', Cruise_line='Azamara', Age=6, Tonnage=30.276999999999997, passengers=6.94, length=
5.94, cabins=3.55, passenger_density=42.64, crew=3.55, cruise_cat=16.0),
 Row(Ship_name='Celebration', Cruise_line='Carnival', Age=26, Tonnage=47.262, passengers=14.86, length=7.2
2, cabins=7.43, passenger_density=31.8, crew=6.7, cruise_cat=1.0),
 Row(Ship_name='Conquest', Cruise_line='Carnival', Age=11, Tonnage=110.0, passengers=29.74, length=9.53, c
abins=14.88, passenger_density=36.99, crew=19.1, cruise_cat=1.0),
 Row(Ship_name='Destiny', Cruise_line='Carnival', Age=17, Tonnage=101.353, passengers=26.42, length=8.92,
```

```
cabins=13.21, passenger_density=38.36, crew=10.0, cruise_cat=1.0)]
```

In [9]:
```
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
```

In [10]:
```
indexed.columns
```

Out[10]:
```
['Ship_name',
 'Cruise_line',
 'Age',
 'Tonnage',
 'passengers',
 'length',
 'cabins',
 'passenger_density',
 'crew',
 'cruise_cat']
```

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

In [12]:
```
output = assembler.transform(indexed)
```

In [13]:
```
output.select("features", "crew").show()
```

```
+--------------------+----+
|            features|crew|
+--------------------+----+
|[6.0,30.276999999...|3.55|
|[6.0,30.276999999...|3.55|
|[26.0,47.262,14.8...| 6.7|
|[11.0,110.0,29.74...|19.1|
|[17.0,101.353,26....|10.0|
|[22.0,70.367,20.5...| 9.2|
|[15.0,70.367,20.5...| 9.2|
|[23.0,70.367,20.5...| 9.2|
|[19.0,70.367,20.5...| 9.2|
|[6.0,110.23899999...|11.5|
|[10.0,110.0,29.74...|11.6|
|[28.0,46.052,14.5...| 6.6|
|[18.0,70.367,20.5...| 9.2|
|[17.0,70.367,20.5...| 9.2|
|[11.0,86.0,21.24,...| 9.3|
|[8.0,110.0,29.74,...|11.6|
|[9.0,88.5,21.24,9...|10.3|
|[15.0,70.367,20.5...| 9.2|
|[12.0,88.5,21.24,...| 9.3|
|[20.0,70.367,20.5...| 9.2|
+--------------------+----+
only showing top 20 rows
```

In [14]:
```
final_data = output.select("features", "crew")
```

In [15]:
```
train_data,test_data = final_data.randomSplit([0.7,0.3])
```

In [16]:
```
from pyspark.ml.regression import LinearRegression
# Create a Linear Regression Model object
lr = LinearRegression(labelCol='crew')
```

In [17]:
```
# Fit the model to the data and call this model lrModel
lrModel = lr.fit(train_data)
```

In [18]:
```
# Print the coefficients and intercept for linear regression
print("Coefficients: {} Intercept: {}".format(lrModel.coefficients,lrModel.intercept))
```

```
Coefficients: [-0.0145423814068,0.0137445818936,-0.111000735058,0.422234330769,0.705574105078,-0.006312026
48669,0.0306212943631] Intercept: -0.5598623529951635
```

In [19]:
```
test_results = lrModel.evaluate(test_data)
```

In [20]:
```
print("RMSE: {}".format(test_results.rootMeanSquaredError))
print("MSE: {}".format(test_results.meanSquaredError))
print("R2: {}".format(test_results.r2))
```

```
RMSE: 1.3174339720092743
MSE: 1.7356322706041332
R2: 0.8671622449217978
```

In [21]:
```
# R2 of 0.86 is pretty good, let's check the data a little closer
from pyspark.sql.functions import corr
```

In [22]:
```
df.select(corr('crew','passengers')).show()
```

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

In [23]:
```
df.select(corr('crew','cabins')).show()
```

```
+------------------+
|corr(crew, cabins)|
+------------------+
```