**Linear Regression Project**

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:



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!

**from** **pyspark.sql** **import** SparkSession

spark = SparkSession.builder.appName('cruise').getOrCreate()

df = spark.read.csv('cruise\_ship\_info.csv',inferSchema=**True**,header=**True**)

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)

df.show()

df.describe().show()

## 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!

df.groupBy('Cruise\_line').count().show()

**from** **pyspark.ml.feature** **import** StringIndexer

indexer = StringIndexer(inputCol="Cruise\_line", outputCol="cruise\_cat")

indexed = indexer.fit(df).transform(df)

indexed.head(5)

**from** **pyspark.ml.linalg** **import** Vectors

**from** **pyspark.ml.feature** **import** VectorAssembler

indexed.columns

assembler = VectorAssembler(

inputCols=['Age',

'Tonnage',

'passengers',

'length',

'cabins',

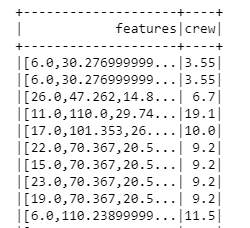
'passenger\_density',

'cruise\_cat'],

outputCol="features")

output = assembler.transform(indexed)

output.select("features", "crew").show()



final\_data = output.select("features", "crew")

train\_data,test\_data = final\_data.randomSplit([0.7,0.3])

**from** **pyspark.ml.regression** **import** LinearRegression

*# Create a Linear Regression Model object*

lr = LinearRegression(labelCol='crew')

*# Fit the model to the data and call this model lrModel*

lrModel = lr.fit(train\_data)

*# Print the coefficients and intercept for linear regression* print("Coefficients: **{}** Intercept: **{}**".format(lrModel.coefficients,lrModel.intercept))

test\_results = lrModel.evaluate(test\_data)

print("RMSE: **{}**".format(test\_results.rootMeanSquaredError))

print("MSE: **{}**".format(test\_results.meanSquaredError))

print("R2: **{}**".format(test\_results.r2))

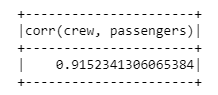
RMSE: 1.3174339720092743

MSE: 1.7356322706041332

R2: 0.8671622449217978

*# R2 of 0.86 is pretty good, let's check the data a little closer* **from** **pyspark.sql.functions** **import** corr

df.select(corr('crew','passengers')).show()



df.select(corr('crew','cabins')).show()

