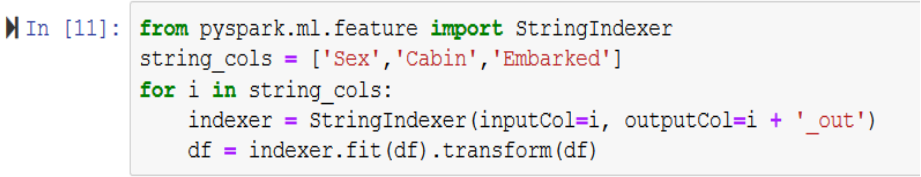
Goal:

- To improve results on an analytical problem I dealt with at the beginning of the previous semester
- Implement suitable machine learning models to predict whether a Titanic passenger was likely to survive given information about where they were, what class they were in, their age, etc.

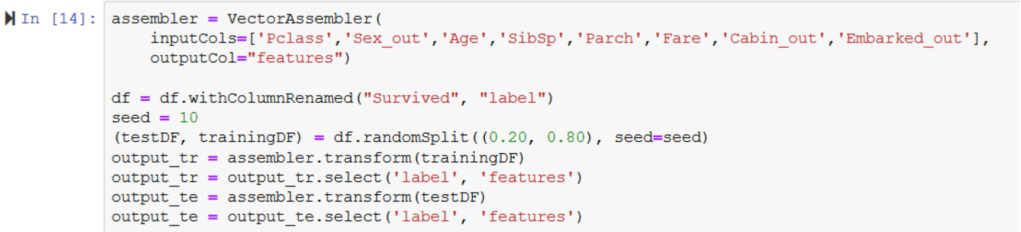
Machine Learning:

- Before analysis could start datasets needed to be modified including label encoding:



```
In [11]:
from pyspark.ml.feature import StringIndexer
string_cols = ['Sex','Cabin','Embarked']
for i in string_cols:
    indexer = StringIndexer(inputCol=i, outputCol=i + '_out')
    df = indexer.fit(df).transform(df)
```

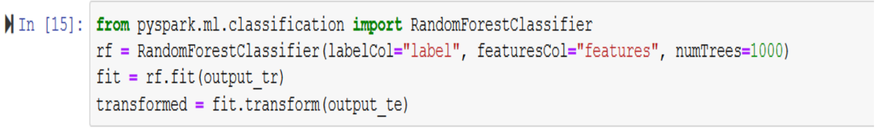
- Vectorizing the numerical features into a sparse vector using Vector Assembler:



```
In [14]:
assembler = VectorAssembler(
    inputCols=['Pclass','Sex_out','Age','SibSp','Parch','Fare','Cabin_out','Embarked_out'],
    outputCol="features")

df = df.withColumnRenamed("Survived", "label")
seed = 10
(testDF, trainingDF) = df.randomSplit((0.20, 0.80), seed=seed)
output_tr = assembler.transform(trainingDF)
output_tr = output_tr.select('label', 'features')
output_te = assembler.transform(testDF)
output_te = output_te.select('label', 'features')
```

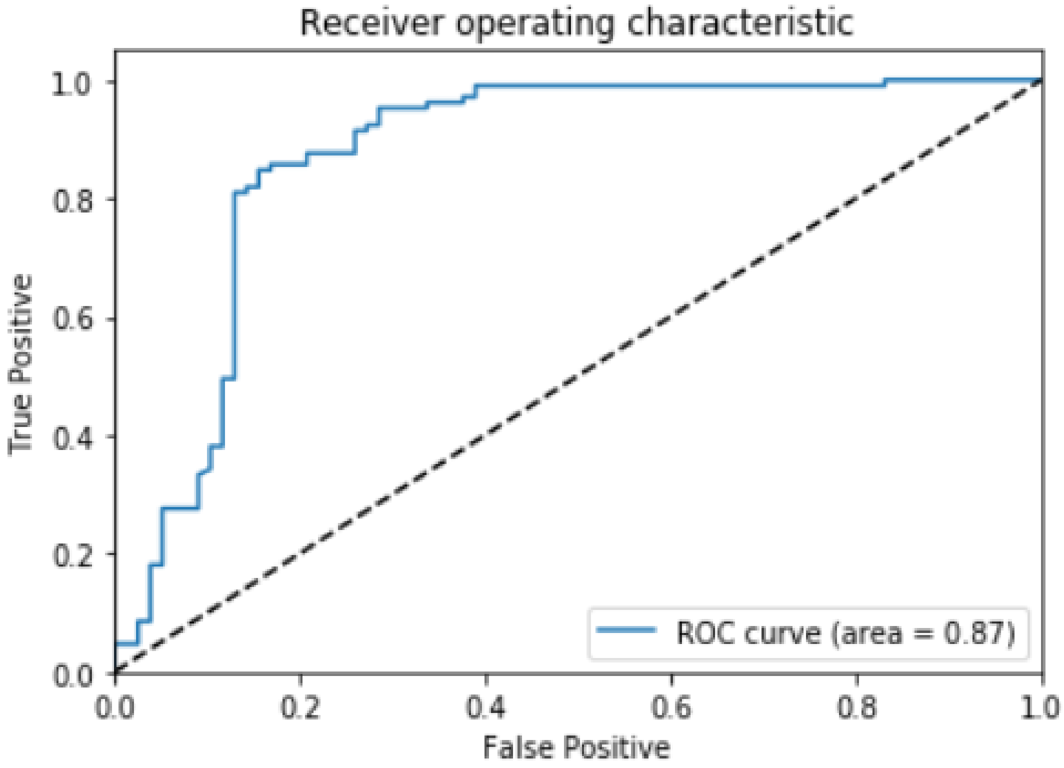
- Fitting the vectorized data with a random forest:



```
In [15]:
from pyspark.ml.classification import RandomForestClassifier
rf = RandomForestClassifier(labelCol="label", featuresCol="features", numTrees=1000)
fit = rf.fit(output_tr)
transformed = fit.transform(output_te)
```

Visualization:

- Following the analysis it was then necessary to visualize how accurate the results were. For this an ROC curve was graphed:


Receiver operating characteristic
True Positive
False Positive
ROC curve (area = 0.87)