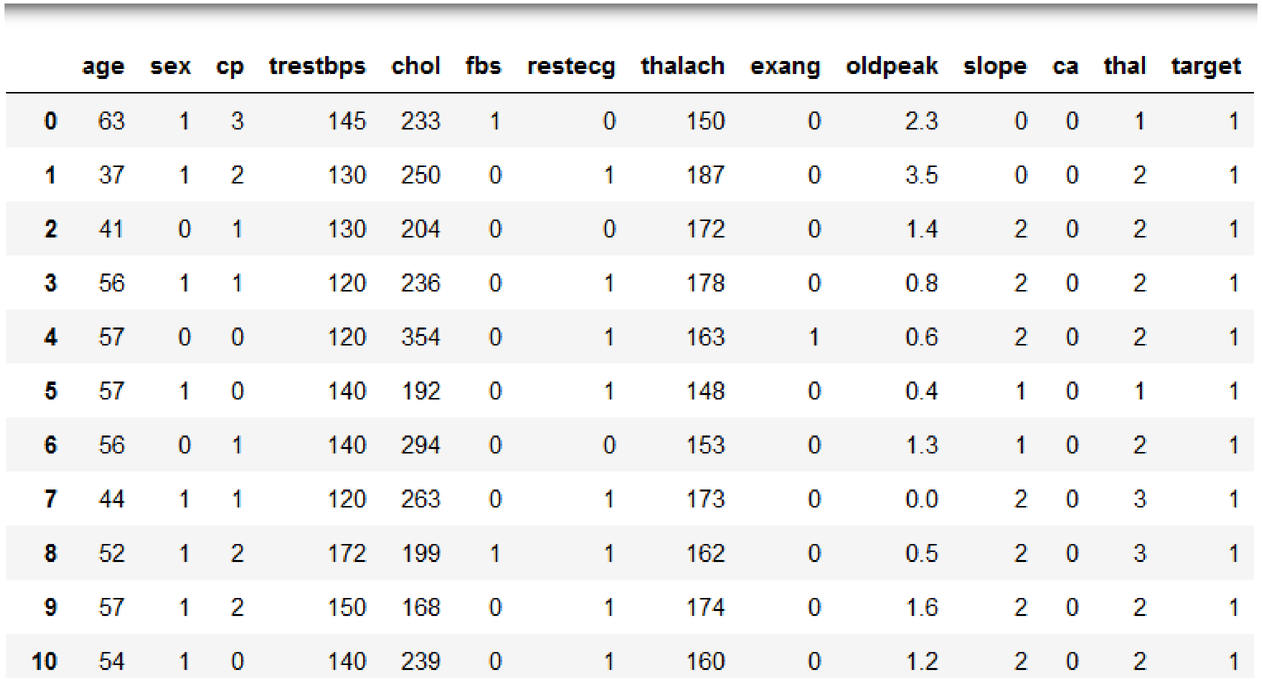
MOTIVATION

- Heart Disease dataset (collected at Cleveland, USA)
- Objective is to predict whether a patient is at risk of heart disease based on certain diagnostic features.
- Dataset has 13 such features like age, sex, cholesterol level, blood pressure , max heart rate, etc.
- Data available for 303 patients.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
10	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1

UNIVERSITY OF VIRGINIA
DATA SCIENCE
INSTITUTE

CODE SNIPPETS

- Creating Spark Context and reading/writing in parquet format

Creating requisite contexts to connect to spark

```
config = pyspark.SparkConf().setAppName('odl').setMaster('local')
spcon = pyspark.SparkContext(conf=config)
sqlcon = pyspark.sql.SQLContext(spcon)
spcon
```

SparkContext

Spark UI

Version

`v2.2.1`

Master

`local`

AppName

`odl`

Writing the heart data to a parquet path

```
parquetPath = '/home/ec2-user/SageMaker/as3uj/heart_pqt'
heart_df.write.parquet(parquetPath)
```

Write to spark dataframe from parquet

```
heart_spdf = sqlcon.read.parquet(parquetPath)
heart_spdf
```

```
heart_train.describe('target').show()
```

```
+-------+------------------+
|summary|            target|
+-------+------------------+
|  count|               245|
|   mean|0.5469387755102041|
| stddev|0.4988108978460373|
|    min|                 0|
|    max|                 1|
+-------+------------------+
```

- Using VectorAssembler for multiple features
- Dividing data into train and test sets with balanced class distribution

Vectorization

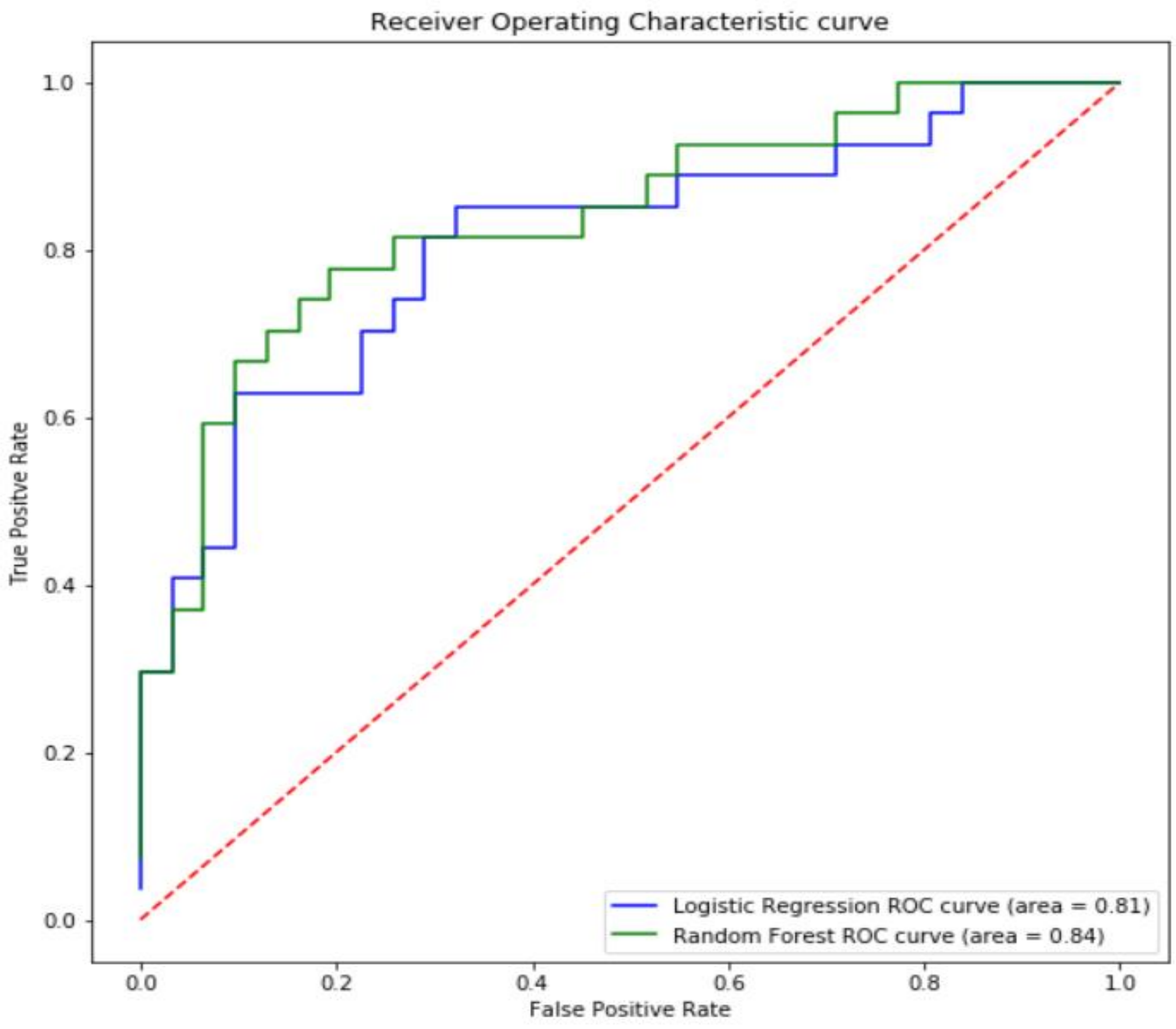
```
ignore = ['target']
assembler = VectorAssembler(
    inputCols=[x for x in heart_train.columns if x not in ignore],
    outputCol='features')
```

```
heart_train = assembler.transform(heart_train)
```

```
heart_test.describe('target').show()
```

```
+-------+------------------+
|summary|            target|
+-------+------------------+
|  count|                58|
|   mean|0.5344827586206896|
| stddev|0.5031660198753178|
|    min|                 0|
|    max|                 1|
+-------+------------------+
```

VISUALIZATION



Evaluation (AUC) for logistic regression

```
evaluator = BinaryClassificationEvaluator
evaluator.evaluate(predictions)
```

```
0.8088410991636799
```

Evaluation (AUC) for random forest

```
evaluator_rf = BinaryClassificationEvaluator()
evaluator_rf.evaluate(predictions_rf)
```

```
0.8375149342891279
```

UNIVERSITY OF VIRGINIA DATA SCIENCE INSTITUTE