ML with PySpark

- Classify/Predict

Datasource

- https://archive.ics.uci.edu/ml/datasets/HCV+data

```
# Load our Pkgs
from pyspark import SparkContext
```

```
sc = SparkContext(master='local[2]')
```

```
# Spark UI
sc
```

SparkContext

Spark UI

Version
    v3.5.3
Master
    local[2]
AppName
    pyspark-shell

```
# Load Pkgs
from pyspark.sql import SparkSession
```

```
# Spark
spark = SparkSession.builder.appName("MLwithSpark").getOrCreate()
```

WorkFlow

- Data Prep
- Feature Engineering
- Build Model
- Evaluate

Task

- Predict if a patient is Hep or not based parameter
- The data set contains laboratory values of blood donors and Hepatitis C patients and demographic values like age.

```
# Load our dataset
df = spark.read.csv("data/hcvdata.csv",header=True,inferSchema=True)
```

```
# Preview Dataset
df.show()
```

```
+---+-------------+---+---+----+----+----+----+----+-----+----+-----+----+----+
|_c0|     Category|Age|Sex| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|
+---+-------------+---+---+----+----+----+----+----+-----+----+-----+----+----+
|  1|0=Blood Donor| 32|  m|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|  69|
|  2|0=Blood Donor| 32|  m|38.5|70.3|  18|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|
|  3|0=Blood Donor| 32|  m|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|
|  4|0=Blood Donor| 32|  m|43.2|  52|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|
|  5|0=Blood Donor| 32|  m|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|
|  6|0=Blood Donor| 32|  m|41.6|43.3|18.5|19.7|12.3| 9.92|6.05|111.0|91.0|  74|
|  7|0=Blood Donor| 32|  m|46.3|41.3|17.5|17.8| 8.5| 7.01|4.79| 70.0|16.9|74.5|
|  8|0=Blood Donor| 32|  m|42.2|41.9|35.8|31.1|16.1| 5.82| 4.6|109.0|21.5|67.1|
|  9|0=Blood Donor| 32|  m|50.9|65.5|23.2|21.2| 6.9| 8.69| 4.1| 83.0|13.7|71.3|
| 10|0=Blood Donor| 32|  m|42.4|86.3|20.3|20.0|35.2| 5.46|4.45| 81.0|15.9|69.9|
| 11|0=Blood Donor| 32|  m|44.3|52.3|21.7|22.4|17.2| 4.15|3.57| 78.0|24.1|75.4|
| 12|0=Blood Donor| 33|  m|46.4|68.2|10.3|20.0| 5.7| 7.36| 4.3| 79.0|18.7|68.6|
| 13|0=Blood Donor| 33|  m|36.3|78.6|23.6|22.0| 7.0| 8.56|5.38| 78.0|19.4|68.7|
| 14|0=Blood Donor| 33|  m|  39|51.7|15.9|24.0| 6.8| 6.46|3.38| 65.0| 7.0|70.4|
| 15|0=Blood Donor| 33|  m|38.7|39.8|22.5|23.0| 4.1| 4.63|4.97| 63.0|15.2|71.9|
| 16|0=Blood Donor| 33|  m|41.8|  65|33.1|38.0| 6.6| 8.83|4.43| 71.0|24.0|72.7|
| 17|0=Blood Donor| 33|  m|40.9|  73|17.2|22.9|10.0| 6.98|5.22| 90.0|14.7|72.4|
| 18|0=Blood Donor| 33|  m|45.2|88.3|32.4|31.2|10.1| 9.78|5.51|102.0|48.5|76.5|
| 19|0=Blood Donor| 33|  m|36.6|57.1|38.9|40.3|24.9| 9.62| 5.5|112.0|27.6|69.3|
| 20|0=Blood Donor| 33|  m|  42|63.1|32.6|34.9|11.2| 7.01|4.05|105.0|19.1|68.1|
+---+-------------+---+---+----+----+----+----+----+-----+----+-----+----+----+
only showing top 20 rows
```

```
# check for columns
print(df.columns)
```

```
['_c0', 'Category', 'Age', 'Sex', 'ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT']
```

```
# Rearrange
df = df.select('Age', 'Sex', 'ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT','Category')
```

```
df.show(5)
```

```
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
|Age|Sex| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|     Category|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
| 32|  m|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|  69|0=Blood Donor|
| 32|  m|38.5|70.3|  18|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|0=Blood Donor|
| 32|  m|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|0=Blood Donor|
| 32|  m|43.2|  52|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|0=Blood Donor|
| 32|  m|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|0=Blood Donor|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
only showing top 5 rows
```

```
# Check for datatypes
# Before InferSchema=True
df.dtypes
```

```
[('Age', 'int'),
 ('Sex', 'string'),
 ('ALB', 'string'),
 ('ALP', 'string'),
 ('ALT', 'string'),
 ('AST', 'double'),
 ('BIL', 'double'),
 ('CHE', 'double'),
 ('CHOL', 'string'),
 ('CREA', 'double'),
 ('GGT', 'double'),
 ('PROT', 'string'),
 ('Category', 'string')]
```

```
# After InferSchema
df.dtypes
```

```
[('Age', 'int'),
 ('Sex', 'string'),
 ('ALB', 'string'),
 ('ALP', 'string'),
 ('ALT', 'string'),
 ('AST', 'double'),
 ('BIL', 'double'),
 ('CHE', 'double'),
 ('CHOL', 'string'),
 ('CREA', 'double'),
 ('GGT', 'double'),
 ('PROT', 'string'),
 ('Category', 'string')]
```

```
# Check for the Schema
df.printSchema()
```

```
root
 |-- Age: integer (nullable = true)
 |-- Sex: string (nullable = true)
 |-- ALB: string (nullable = true)
 |-- ALP: string (nullable = true)
 |-- ALT: string (nullable = true)
 |-- AST: double (nullable = true)
 |-- BIL: double (nullable = true)
 |-- CHE: double (nullable = true)
 |-- CHOL: string (nullable = true)
 |-- CREA: double (nullable = true)
 |-- GGT: double (nullable = true)
 |-- PROT: string (nullable = true)
 |-- Category: string (nullable = true)
```

```
# Descriptive summary
print(df.describe().show())
```

```
+-------+------------------+----+-----------------+------------------+------------------+-----------------+------------------+----------
|summary|               Age| Sex|              ALB|               ALP|               ALT|              AST|               BIL|
+-------+------------------+----+-----------------+------------------+------------------+-----------------+------------------+----------
|  count|               615| 615|              615|               615|               615|              615|               615|
|   mean| 47.40813008130081|NULL|41.62019543973941| 68.28391959798999| 28.45081433224754|34.78634146341462|11.396747967479675| 8.1966341
| stddev|10.055105445519239|NULL|5.780629404103076|26.028315300123676|25.469688813870942|33.09069033855156|19.673149805846588|2.20565727
|    min|                19|   f|             14.9|             100.4|               0.9|             10.6|               0.8|
|    max|                77|   m|               NA|                NA|                NA|            324.0|             254.0|
+-------+------------------+----+-----------------+------------------+------------------+-----------------+------------------+----------

None
```

```
# Value Count
df.groupBy('Category').count().show()
```

```
+--------------------+-----+
|            Category|count|
+--------------------+-----+
|       0=Blood Donor|  533|
|         3=Cirrhosis|   30|
|          2=Fibrosis|   21|
|0s=suspect Blood ...|    7|
|         1=Hepatitis|   24|
+--------------------+-----+
```

Feature Engineering

- Numerical Values
- Vectorization
- Scaling

```
df.show(5)
```

```
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
|Age|Sex| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|     Category|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
| 32|  m|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|  69|0=Blood Donor|
| 32|  m|38.5|70.3|  18|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|0=Blood Donor|
| 32|  m|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|0=Blood Donor|
| 32|  m|43.2|  52|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|0=Blood Donor|
| 32|  m|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|0=Blood Donor|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
only showing top 5 rows
```

```
import pyspark.ml
```

```
dir(pyspark.ml)
```

```
['Estimator',
 'Model',
 'Pipeline',
 'PipelineModel',
 'PredictionModel',
 'Predictor',
 'TorchDistributor',
 'Transformer',
 'UnaryTransformer',
 '__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 'base',
 'classification',
 'clustering',
 'common',
 'evaluation',
 'feature',
 'fpm',
 'image',
 'linalg',
 'param',
 'pipeline',
 'recommendation',
 'regression',
 'stat',
 'torch',
 'tree',
 'tuning',
 'util',
 'wrapper']
```

```
# Load ML Pkgs
from pyspark.ml.feature import VectorAssembler,StringIndexer
```

```
df.show(4)
```

```
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
|Age|Sex| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|     Category|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
| 32|  m|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|  69|0=Blood Donor|
| 32|  m|38.5|70.3|  18|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|0=Blood Donor|
| 32|  m|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|0=Blood Donor|
| 32|  m|43.2|  52|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|0=Blood Donor|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+
only showing top 4 rows
```

```
# Unique Values for Sex
df.select('Sex').distinct().show()
```

```
+---+
|Sex|
+---+
|  m|
|  f|
+---+
```

```
# Convert the string into numerical code
# label encoding
genderEncoder = StringIndexer(inputCol='Sex',outputCol='Gender').fit(df)
```

```
df = genderEncoder.transform(df)
```

```
df.show(5)
```

```
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+
|Age|Sex| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|     Category|Gender|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+
| 32|  m|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|  69|0=Blood Donor|   0.0|
| 32|  m|38.5|70.3|  18|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|0=Blood Donor|   0.0|
| 32|  m|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|0=Blood Donor|   0.0|
| 32|  m|43.2|  52|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|0=Blood Donor|   0.0|
| 32|  m|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|0=Blood Donor|   0.0|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+
only showing top 5 rows
```

```
# Encoding for Category
# Label Encoding
catEncoder = StringIndexer(inputCol='Category',outputCol='Target').fit(df)
df = catEncoder.transform(df)
```

```
df.show(5)
```

```
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+
|Age|Sex| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|     Category|Gender|Target|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+
| 32|  m|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|  69|0=Blood Donor|   0.0|   0.0|
| 32|  m|38.5|70.3|  18|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|0=Blood Donor|   0.0|   0.0|
| 32|  m|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|0=Blood Donor|   0.0|   0.0|
| 32|  m|43.2|  52|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|0=Blood Donor|   0.0|   0.0|
| 32|  m|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|0=Blood Donor|   0.0|   0.0|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+
only showing top 5 rows
```

```
# Get the labels
catEncoder.labels
```

```
['0=Blood Donor',
 '3=Cirrhosis',
 '1=Hepatitis',
 '2=Fibrosis',
 '0s=suspect Blood Donor']
```

```
# IndexToString
from pyspark.ml.feature import IndexToString
```

```
converter = IndexToString(inputCol='Target',outputCol='orig_cat')
```

```
converted_df = converter.transform(df)
```

```
converted_df.show()
```

```
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+-------------+
|Age|Sex| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|     Category|Gender|Target|     orig_cat|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+-------------+
| 32|  m|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|  69|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|38.5|70.3|  18|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|43.2|  52|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|41.6|43.3|18.5|19.7|12.3| 9.92|6.05|111.0|91.0|  74|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|46.3|41.3|17.5|17.8| 8.5| 7.01|4.79| 70.0|16.9|74.5|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|42.2|41.9|35.8|31.1|16.1| 5.82| 4.6|109.0|21.5|67.1|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|50.9|65.5|23.2|21.2| 6.9| 8.69| 4.1| 83.0|13.7|71.3|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|42.4|86.3|20.3|20.0|35.2| 5.46|4.45| 81.0|15.9|69.9|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 32|  m|44.3|52.3|21.7|22.4|17.2| 4.15|3.57| 78.0|24.1|75.4|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|46.4|68.2|10.3|20.0| 5.7| 7.36| 4.3| 79.0|18.7|68.6|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|36.3|78.6|23.6|22.0| 7.0| 8.56|5.38| 78.0|19.4|68.7|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|  39|51.7|15.9|24.0| 6.8| 6.46|3.38| 65.0| 7.0|70.4|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|38.7|39.8|22.5|23.0| 4.1| 4.63|4.97| 63.0|15.2|71.9|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|41.8|  65|33.1|38.0| 6.6| 8.83|4.43| 71.0|24.0|72.7|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|40.9|  73|17.2|22.9|10.0| 6.98|5.22| 90.0|14.7|72.4|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|45.2|88.3|32.4|31.2|10.1| 9.78|5.51|102.0|48.5|76.5|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|36.6|57.1|38.9|40.3|24.9| 9.62| 5.5|112.0|27.6|69.3|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
| 33|  m|  42|63.1|32.6|34.9|11.2| 7.01|4.05|105.0|19.1|68.1|0=Blood Donor|   0.0|   0.0|0=Blood Donor|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+-------------+
only showing top 20 rows
```

```
### Feature
df.show()
```

```
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+
|Age|Sex| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|     Category|Gender|Target|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+
| 32|  m|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|  69|0=Blood Donor|   0.0|   0.0|
| 32|  m|38.5|70.3|  18|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|0=Blood Donor|   0.0|   0.0|
| 32|  m|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|0=Blood Donor|   0.0|   0.0|
| 32|  m|43.2|  52|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|0=Blood Donor|   0.0|   0.0|
| 32|  m|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|0=Blood Donor|   0.0|   0.0|
| 32|  m|41.6|43.3|18.5|19.7|12.3| 9.92|6.05|111.0|91.0|  74|0=Blood Donor|   0.0|   0.0|
| 32|  m|46.3|41.3|17.5|17.8| 8.5| 7.01|4.79| 70.0|16.9|74.5|0=Blood Donor|   0.0|   0.0|
| 32|  m|42.2|41.9|35.8|31.1|16.1| 5.82| 4.6|109.0|21.5|67.1|0=Blood Donor|   0.0|   0.0|
| 32|  m|50.9|65.5|23.2|21.2| 6.9| 8.69| 4.1| 83.0|13.7|71.3|0=Blood Donor|   0.0|   0.0|
| 32|  m|42.4|86.3|20.3|20.0|35.2| 5.46|4.45| 81.0|15.9|69.9|0=Blood Donor|   0.0|   0.0|
| 32|  m|44.3|52.3|21.7|22.4|17.2| 4.15|3.57| 78.0|24.1|75.4|0=Blood Donor|   0.0|   0.0|
| 33|  m|46.4|68.2|10.3|20.0| 5.7| 7.36| 4.3| 79.0|18.7|68.6|0=Blood Donor|   0.0|   0.0|
| 33|  m|36.3|78.6|23.6|22.0| 7.0| 8.56|5.38| 78.0|19.4|68.7|0=Blood Donor|   0.0|   0.0|
| 33|  m|  39|51.7|15.9|24.0| 6.8| 6.46|3.38| 65.0| 7.0|70.4|0=Blood Donor|   0.0|   0.0|
| 33|  m|38.7|39.8|22.5|23.0| 4.1| 4.63|4.97| 63.0|15.2|71.9|0=Blood Donor|   0.0|   0.0|
| 33|  m|41.8|  65|33.1|38.0| 6.6| 8.83|4.43| 71.0|24.0|72.7|0=Blood Donor|   0.0|   0.0|
| 33|  m|40.9|  73|17.2|22.9|10.0| 6.98|5.22| 90.0|14.7|72.4|0=Blood Donor|   0.0|   0.0|
| 33|  m|45.2|88.3|32.4|31.2|10.1| 9.78|5.51|102.0|48.5|76.5|0=Blood Donor|   0.0|   0.0|
| 33|  m|36.6|57.1|38.9|40.3|24.9| 9.62| 5.5|112.0|27.6|69.3|0=Blood Donor|   0.0|   0.0|
| 33|  m|  42|63.1|32.6|34.9|11.2| 7.01|4.05|105.0|19.1|68.1|0=Blood Donor|   0.0|   0.0|
+---+---+----+----+----+----+----+-----+----+-----+----+----+-------------+------+------+
only showing top 20 rows
```

```
print(df.columns)
```

```
['Age', 'Sex', 'ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT', 'Category', 'Gender', 'Target']
```

```
df.dtypes
```

```
[('Age', 'int'),
 ('Sex', 'string'),
 ('ALB', 'string'),
 ('ALP', 'string'),
 ('ALT', 'string'),
 ('AST', 'double'),
 ('BIL', 'double'),
 ('CHE', 'double'),
 ('CHOL', 'string'),
 ('CREA', 'double'),
 ('GGT', 'double'),
 ('PROT', 'string'),
 ('Category', 'string'),
 ('Gender', 'double'),
 ('Target', 'double')]
```

```
df2 = df.select('Age','Gender', 'ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT', 'Target')
```

```
df2.printSchema()
```

```
root
 |-- Age: integer (nullable = true)
 |-- Gender: double (nullable = false)
 |-- ALB: string (nullable = true)
 |-- ALP: string (nullable = true)
 |-- ALT: string (nullable = true)
 |-- AST: double (nullable = true)
 |-- BIL: double (nullable = true)
 |-- CHE: double (nullable = true)
 |-- CHOL: string (nullable = true)
 |-- CREA: double (nullable = true)
 |-- GGT: double (nullable = true)
 |-- PROT: string (nullable = true)
 |-- Target: double (nullable = false)
```

```
# df2.fillna(0,subset=['col1'])
```

```
df2 = df2.toPandas().replace('NA',0).astype(float)
```

```
type(df2)
```

```
pandas.core.frame.DataFrame
def __init__(data=None, index: Axes | None=None, columns: Axes | None=None, dtype: Dtype |
None=None, copy: bool | None=None) -> None

/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py
Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns).
Arithmetic operations align on both row and column labels. Can be
thought of as a dict-like container for Series objects. The primary
```

```
type(df)
```

```
pyspark.sql.dataframe.DataFrame
def __init__(jdf: JavaObject, sql_ctx: Union['SQLContext', 'SparkSession'])

/usr/local/lib/python3.10/dist-packages/pyspark/sql/dataframe.py
A distributed collection of data grouped into named columns.

.. versionadded:: 1.3.0

   versionchanged:: 3.4.0
```

```
# Convert To PySpark Dataframe
new_df = spark.createDataFrame(df2)
```

```
new_df.show()
```

```
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+
| Age|Gender| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|Target|
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+
|32.0|   0.0|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|69.0|   0.0|
|32.0|   0.0|38.5|70.3|18.0|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|   0.0|
|32.0|   0.0|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|   0.0|
|32.0|   0.0|43.2|52.0|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|   0.0|
|32.0|   0.0|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|   0.0|
|32.0|   0.0|41.6|43.3|18.5|19.7|12.3| 9.92|6.05|111.0|91.0|74.0|   0.0|
|32.0|   0.0|46.3|41.3|17.5|17.8| 8.5| 7.01|4.79| 70.0|16.9|74.5|   0.0|
|32.0|   0.0|42.2|41.9|35.8|31.1|16.1| 5.82| 4.6|109.0|21.5|67.1|   0.0|
|32.0|   0.0|50.9|65.5|23.2|21.2| 6.9| 8.69| 4.1| 83.0|13.7|71.3|   0.0|
|32.0|   0.0|42.4|86.3|20.3|20.0|35.2| 5.46|4.45| 81.0|15.9|69.9|   0.0|
|32.0|   0.0|44.3|52.3|21.7|22.4|17.2| 4.15|3.57| 78.0|24.1|75.4|   0.0|
|33.0|   0.0|46.4|68.2|10.3|20.0| 5.7| 7.36| 4.3| 79.0|18.7|68.6|   0.0|
|33.0|   0.0|36.3|78.6|23.6|22.0| 7.0| 8.56|5.38| 78.0|19.4|68.7|   0.0|
|33.0|   0.0|39.0|51.7|15.9|24.0| 6.8| 6.46|3.38| 65.0| 7.0|70.4|   0.0|
|33.0|   0.0|38.7|39.8|22.5|23.0| 4.1| 4.63|4.97| 63.0|15.2|71.9|   0.0|
|33.0|   0.0|41.8|65.0|33.1|38.0| 6.6| 8.83|4.43| 71.0|24.0|72.7|   0.0|
|33.0|   0.0|40.9|73.0|17.2|22.9|10.0| 6.98|5.22| 90.0|14.7|72.4|   0.0|
|33.0|   0.0|45.2|88.3|32.4|31.2|10.1| 9.78|5.51|102.0|48.5|76.5|   0.0|
|33.0|   0.0|36.6|57.1|38.9|40.3|24.9| 9.62| 5.5|112.0|27.6|69.3|   0.0|
|33.0|   0.0|42.0|63.1|32.6|34.9|11.2| 7.01|4.05|105.0|19.1|68.1|   0.0|
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+
only showing top 20 rows
```

```
# Check For DTYpes and Schema
new_df.printSchema()
```

```
root
 |-- Age: double (nullable = true)
 |-- Gender: double (nullable = true)
 |-- ALB: double (nullable = true)
 |-- ALP: double (nullable = true)
 |-- ALT: double (nullable = true)
 |-- AST: double (nullable = true)
 |-- BIL: double (nullable = true)
 |-- CHE: double (nullable = true)
 |-- CHOL: double (nullable = true)
 |-- CREA: double (nullable = true)
 |-- GGT: double (nullable = true)
 |-- PROT: double (nullable = true)
 |-- Target: double (nullable = true)
```

```
required_features = ['Age','Gender', 'ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT', 'Target']
```

```
# VectorAsm
vec_assembler = VectorAssembler(inputCols=required_features,outputCol='features')
```

```
vec_df = vec_assembler.transform(new_df)
```

```
vec_df.show(5)
```

```
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+--------------------+
| Age|Gender| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|Target|            features|
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+--------------------+
|32.0|   0.0|38.5|52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|69.0|   0.0|[32.0,0.0,38.5,52...|
|32.0|   0.0|38.5|70.3|18.0|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|   0.0|[32.0,0.0,38.5,70...|
|32.0|   0.0|46.9|74.7|36.2|52.6| 6.1| 8.84| 5.2| 86.0|33.2|79.3|   0.0|[32.0,0.0,46.9,74...|
|32.0|   0.0|43.2|52.0|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|   0.0|[32.0,0.0,43.2,52...|
|32.0|   0.0|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|   0.0|[32.0,0.0,39.2,74...|
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+--------------------+
only showing top 5 rows
```

Train, Test Split

```
train_df,test_df = vec_df.randomSplit([0.7,0.3])
```

```
train_df.count()
```

```
444
```

```
train_df.show(4)
```

```
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+--------------------+
| Age|Gender| ALB| ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|Target|            features|
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+--------------------+
|32.0|   0.0|38.5|70.3|18.0|24.7| 3.9|11.17| 4.8| 74.0|15.6|76.5|   0.0|[32.0,0.0,38.5,70...|
|32.0|   0.0|39.2|74.1|32.6|24.8| 9.6| 9.15|4.32| 76.0|29.9|68.7|   0.0|[32.0,0.0,39.2,74...|
|32.0|   0.0|41.6|43.3|18.5|19.7|12.3| 9.92|6.05|111.0|91.0|74.0|   0.0|[32.0,0.0,41.6,43...|
|32.0|   0.0|42.2|41.9|35.8|31.1|16.1| 5.82| 4.6|109.0|21.5|67.1|   0.0|[32.0,0.0,42.2,41...|
+----+------+----+----+----+----+----+-----+----+-----+----+----+------+--------------------+
only showing top 4 rows
```

Model Building

- Pyspark.ml: DataFrame
- Pyspark.mllib: RDD /Legacy

```
from pyspark.ml.classification import LogisticRegression,DecisionTreeClassifier
```

```
# Logist Model
lr = LogisticRegression(featuresCol='features',labelCol='Target')
```

```
lr_model = lr.fit(train_df)
```

```
y_pred = lr_model.transform(test_df)
```

```
y_pred.show()
```

```
+----+------+----+-----+----+----+----+-----+----+-----+----+----+------+--------------------+--------------------+--------------------+
| Age|Gender| ALB|  ALP| ALT| AST| BIL|  CHE|CHOL| CREA| GGT|PROT|Target|            features|       rawPrediction|         probability|
+----+------+----+-----+----+----+----+-----+----+-----+----+----+------+--------------------+--------------------+--------------------+
|32.0|   0.0|38.5| 52.5| 7.7|22.1| 7.5| 6.93|3.23|106.0|12.1|69.0|   0.0|[32.0,0.0,38.5,52...|[111.571240149727...|[1.0,2.7712838143...|
|32.0|   0.0|43.2| 52.0|30.6|22.6|18.9| 7.33|4.74| 80.0|33.8|75.7|   0.0|[32.0,0.0,43.2,52...|[97.9975751452358...|[1.0,7.5803653908...|
|32.0|   0.0|46.3| 41.3|17.5|17.8| 8.5| 7.01|4.79| 70.0|16.9|74.5|   0.0|[32.0,0.0,46.3,41...|[110.584371420791...|[1.0,2.0134323979...|
|33.0|   0.0|36.6| 57.1|38.9|40.3|24.9| 9.62| 5.5|112.0|27.6|69.3|   0.0|[33.0,0.0,36.6,57...|[83.6212567532894...|[1.0,5.5977877055...|
|33.0|   0.0|38.7| 39.8|22.5|23.0| 4.1| 4.63|4.97| 63.0|15.2|71.9|   0.0|[33.0,0.0,38.7,39...|[116.675698255542...|[1.0,5.0309907171...|
|33.0|   0.0|39.0| 51.7|15.9|24.0| 6.8| 6.46|3.38| 65.0| 7.0|70.4|   0.0|[33.0,0.0,39.0,51...|[121.438570894806...|[1.0,1.2145807759...|
|33.0|   0.0|40.9| 73.0|17.2|22.9|10.0| 6.98|5.22| 90.0|14.7|72.4|   0.0|[33.0,0.0,40.9,73...|[109.518558760129...|[1.0,5.5851402189...|
|33.0|   0.0|41.8| 65.0|33.1|38.0| 6.6| 8.83|4.43| 71.0|24.0|72.7|   0.0|[33.0,0.0,41.8,65...|[106.211282108152...|[1.0,1.4814326466...|
|33.0|   0.0|46.7| 88.3|23.4|23.9| 7.8| 9.42|4.62| 78.0|29.5|74.3|   0.0|[33.0,0.0,46.7,88...|[112.912860600975...|[1.0,3.5495554687...|
|34.0|   0.0|29.0| 41.6|29.1|16.1| 4.8| 6.82|4.03| 62.0|14.5|53.2|   0.0|[34.0,0.0,29.0,41...|[154.039673939318...|[1.0,4.2715835028...|
|34.0|   0.0|40.5| 32.4|29.6|27.1| 5.8| 10.5|4.56| 91.0|26.6|72.0|   0.0|[34.0,0.0,40.5,32...|[98.3180485553592...|[1.0,1.1912375817...|
|35.0|   0.0|44.5| 70.3|26.2|25.1| 5.1|10.12|4.69| 82.0|20.7|67.2|   0.0|[35.0,0.0,44.5,70...|[122.036434211206...|[1.0,1.1835349803...|
|36.0|   0.0|42.6| 65.3|35.8|27.1|15.7|10.66|4.38| 96.0|34.7|71.0|   0.0|[36.0,0.0,42.6,65...|[102.219225782006...|[1.0,5.2129719370...|
|36.0|   0.0|48.7| 65.0|11.5|18.0| 7.4| 8.02|7.35| 69.0|14.2|73.4|   0.0|[36.0,0.0,48.7,65...|[110.587874910060...|[1.0,3.9255855469...|
|36.0|   0.0|48.9| 82.8|16.9|24.4| 8.9| 8.91| 5.1| 97.0|14.8|79.9|   0.0|[36.0,0.0,48.9,82...|[96.6444562777555...|[1.0,1.2109970278...|
|37.0|   0.0|31.4|106.0|16.6|17.0| 2.4| 5.95| 5.3| 68.0|22.9|72.3|   0.0|[37.0,0.0,31.4,10...|[126.163167032580...|[1.0,2.4987768426...|
|37.0|   0.0|33.9| 64.0|91.7|44.7| 9.1| 8.35| 5.4| 95.0|30.3|74.7|   0.0|[37.0,0.0,33.9,64...|[98.6132901167605...|[1.0,1.1511122787...|
|37.0|   0.0|42.9| 70.7|16.3|24.1|15.7| 9.03| 6.8| 93.0|70.1|73.4|   0.0|[37.0,0.0,42.9,70...|[86.0454111374304...|[1.0,1.3188355850...|
|37.0|   0.0|43.6| 72.8|51.4|43.7|13.8| 8.16|4.88| 70.0|94.5|75.2|   0.0|[37.0,0.0,43.6,72...|[87.9008794902910...|[1.0,1.1095871962...|
|37.0|   0.0|44.0| 57.4|26.1|24.6| 9.7|10.41|6.17| 83.0|38.9|76.5|   0.0|[37.0,0.0,44.0,57...|[91.1836224357164...|[1.0,1.1014304625...|
+----+------+----+-----+----+----+----+-----+----+-----+----+----+------+--------------------+--------------------+--------------------+
only showing top 20 rows
```

```
print(y_pred.columns)
```

```
['Age', 'Gender', 'ALB', 'ALP', 'ALT', 'AST', 'BIL', 'CHE', 'CHOL', 'CREA', 'GGT', 'PROT', 'Target', 'features', 'rawPrediction', 'proba
```

```
y_pred.select('target','rawPrediction', 'probability', 'prediction').show()
```

```
+------+--------------------+--------------------+----------+
|target|       rawPrediction|         probability|prediction|
+------+--------------------+--------------------+----------+
|   0.0|[111.571240149727...|[1.0,2.7712838143...|       0.0|
|   0.0|[97.9975751452358...|[1.0,7.5803653908...|       0.0|
|   0.0|[110.584371420791...|[1.0,2.0134323979...|       0.0|
|   0.0|[83.6212567532894...|[1.0,5.5977877055...|       0.0|
|   0.0|[116.675698255542...|[1.0,5.0309907171...|       0.0|
|   0.0|[121.438570894806...|[1.0,1.2145807759...|       0.0|
|   0.0|[109.518558760129...|[1.0,5.5851402189...|       0.0|
|   0.0|[106.211282108152...|[1.0,1.4814326466...|       0.0|
|   0.0|[112.912860600975...|[1.0,3.5495554687...|       0.0|
|   0.0|[154.039673939318...|[1.0,4.2715835028...|       0.0|
|   0.0|[98.3180485553592...|[1.0,1.1912375817...|       0.0|
|   0.0|[122.036434211206...|[1.0,1.1835349803...|       0.0|
|   0.0|[102.219225782006...|[1.0,5.2129719370...|       0.0|
|   0.0|[110.587874910060...|[1.0,3.9255855469...|       0.0|
|   0.0|[96.6444562777555...|[1.0,1.2109970278...|       0.0|
|   0.0|[126.163167032580...|[1.0,2.4987768426...|       0.0|
|   0.0|[98.6132901167605...|[1.0,1.1511122787...|       0.0|
|   0.0|[86.0454111374304...|[1.0,1.3188355850...|       0.0|
|   0.0|[87.9008794902910...|[1.0,1.1095871962...|       0.0|
|   0.0|[91.1836224357164...|[1.0,1.1014304625...|       0.0|
+------+--------------------+--------------------+----------+
only showing top 20 rows
```

Model Evaluation

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

```
# How to Check For Accuracy
multi_evaluator = MulticlassClassificationEvaluator(labelCol='Target',metricName='accuracy')
```

```
multi_evaluator.evaluate(y_pred)
```

```
0.9590643274853801
```

Precision, F1 Score, Recall : Classification Report

```
from pyspark.mllib.evaluation import MulticlassMetrics
```

```
lr_metric = MulticlassMetrics(y_pred['target', 'prediction'].rdd)
```

```
/usr/local/lib/python3.10/dist-packages/pyspark/sql/context.py:158: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCr
  warnings.warn(
```

```
dir(lr_metric)
```

```
['__class__',
 '__del__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 '_java_model',
 '_sc',
 'accuracy',
 'call',
 'confusionMatrix',
 'fMeasure',
 'falsePositiveRate',
 'logLoss',
 'precision',
 'recall',
 'truePositiveRate',
 'weightedFMeasure',
 'weightedFalsePositiveRate',
 'weightedPrecision',
 'weightedRecall',
 'weightedTruePositiveRate']
```

```
print("Accuracy",lr_metric.accuracy)
```

```
Accuracy 0.9590643274853801
```

```
print("Precision",lr_metric.precision(1.0))
print("Recall",lr_metric.recall(1.0))
print("F1Score",lr_metric.fMeasure(1.0))
```

```
Precision 1.0
Recall 1.0
F1Score 1.0
```

```
dir(lr_model)
```

```
 'probabilityCol',
 'rawPredictionCol',
 'read',
 'regParam',
 'save',
 'set',
 'setFeaturesCol',
 'setPredictionCol',
 'setProbabilityCol',
 'setRawPredictionCol',
 'setThreshold',
 'setThresholds',
 'standardization',
 'summary',
 'threshold',
 'thresholds',
 'tol',
 'transform',
 'uid',
 'upperBoundsOnCoefficients',
 'upperBoundsOnIntercepts',
 'weightCol',
 'write']
```

```
# Saving Model
lr_model.save("lr_model_30")
```

```
lr_model.write().save("mylr_model")
```