PySpark - Diabetes Prediction

In [ ]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('spark').getOrCreate()
```

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

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

```
+-----------+-------+-------------+-------------+-------+----+------------------------
-+---+-------+
|Pregnancies|Glucose|BloodPressure|SkinThickness|Insulin| BMI|DiabetesPedigreeFunctio
n|Age|Outcome|
+-----------+-------+-------------+-------------+-------+----+------------------------
-+---+-------+
|          6|    148|           72|           35|      0|33.6|                   0.62
7| 50|      1|
|          1|     85|           66|           29|      0|26.6|                   0.35
1| 31|      0|
|          8|    183|           64|            0|      0|23.3|                   0.67
2| 32|      1|
|          1|     89|           66|           23|     94|28.1|                   0.16
7| 21|      0|
|          0|    137|           40|           35|    168|43.1|                   2.28
8| 33|      1|
|          5|    116|           74|            0|      0|25.6|                   0.20
1| 30|      0|
|          3|     78|           50|           32|     88|31.0|                   0.24
8| 26|      1|
|         10|    115|            0|            0|      0|35.3|                   0.13
4| 29|      0|
|          2|    197|           70|           45|    543|30.5|                   0.15
8| 53|      1|
|          8|    125|           96|            0|      0| 0.0|                   0.23
2| 54|      1|
|          4|    110|           92|            0|      0|37.6|                   0.19
1| 30|      0|
|         10|    168|           74|            0|      0|38.0|                   0.53
7| 34|      1|
|         10|    139|           80|            0|      0|27.1|                   1.44
1| 57|      0|
|          1|    189|           60|           23|    846|30.1|                   0.39
8| 59|      1|
|          5|    166|           72|           19|    175|25.8|                   0.58
7| 51|      1|
|          7|    100|            0|            0|      0|30.0|                   0.48
4| 32|      1|
|          0|    118|           84|           47|    230|45.8|                   0.55
1| 31|      1|
|          7|    107|           74|            0|      0|29.6|                   0.25
4| 31|      1|
```

```
|          1|    103|           30|           38|     83|43.3|                    0.18
3| 33|      0|
|          1|    115|           70|           30|     96|34.6|                    0.52
9| 32|      1|
+-----------+-------+-------------+-------------+-------+----+--------------------
-+---+-------+
only showing top 20 rows
```

In [ ]:
```
df.printSchema()
```

```
root
 |-- Pregnancies: integer (nullable = true)
 |-- Glucose: integer (nullable = true)
 |-- BloodPressure: integer (nullable = true)
 |-- SkinThickness: integer (nullable = true)
 |-- Insulin: integer (nullable = true)
 |-- BMI: double (nullable = true)
 |-- DiabetesPedigreeFunction: double (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Outcome: integer (nullable = true)
```

In [ ]:
```
print((df.count(), len(df.columns)))
```

```
(768, 9)
```

In [ ]:
```
df.groupby('Outcome').count().show()
```

```
+-------+-----+
|Outcome|count|
+-------+-----+
|      1|  268|
|      0|  500|
+-------+-----+
```

In [ ]:
```
df.describe().show()
```

```
+-------+------------------+-----------------+------------------+------------------+-----------------+-
-----------------+------------------+-----------------------+------------------+----
--------------+
|summary|       Pregnancies|          Glucose|     BloodPressure|     SkinThickness|
Insulin|               BMI|DiabetesPedigreeFunction|               Age|           Out
come|
+-------+------------------+-----------------+------------------+------------------+-----------------+-
-----------------+------------------+-----------------------+------------------+----
--------------+
|  count|               768|              768|               768|               768|
768|               768|                   768|               768|               768
|
|   mean|3.8450520833333335|     120.89453125|       69.10546875|20.536458333333332|
79.79947916666667|31.992578124999977|     0.4718763020833327|33.240885416666664|0.34
89583333333333|
| stddev|  3.36957806269887|31.97261819513622|19.355807170644777|15.952217567727642|1
15.24400235133803| 7.884160320375441|      0.331328595012775|11.760231540678689| 0.4
76951377242799|
|    min|                 0|                0|                 0|                 0|
0|               0.0|                  0.078|                21|                 0|
|    max|                17|              199|               122|                99|
846|              67.1|                   2.42|                81|                 1
|
+-------+------------------+-----------------+------------------+------------------+-----------------+-
-----------------+------------------+-----------------------+------------------+----
--------------+
```

Data Cleaning & Preparation

In [ ]:
```
for col in df.columns:
  print(col + ":" , df[df[col].isNull()].count())
```

```
Pregnancies: 0
Glucose: 0
BloodPressure: 0
SkinThickness: 0
Insulin: 0
BMI: 0
DiabetesPedigreeFunction: 0
Age: 0
Outcome: 0
```

In [ ]:
```
def count_zeros():
  columns_list  = ['Glucose','BloodPressure','SkinThickness', 'Insulin','BMI']
  for col in columns_list:
    print(col + ':', df[df[col]==0].count() )
```

In [ ]:
```
count_zeros()
```

```
Glucose: 5
BloodPressure: 35
SkinThickness: 227
Insulin: 374
BMI: 11
```

In [ ]:
```
from pyspark.sql.functions import *
```

In [ ]:
```
df.agg({'BMI':'mean'}).first()[0]
```

Out[ ]:
```
31.992578124999977
```

In [ ]:
```
for col in df.columns[1:6]:
  data = df.agg({col:'mean'}).first()[0]
  print(f'Mean value for {col} is {int(data)}')
  df = df.withColumn(col, when(df[col]== 0, int(data)).otherwise(df[col]))
```

```
Mean value for Glucose is 120
Mean value for BloodPressure is 69
Mean value for SkinThickness is 20
Mean value for Insulin is 79
Mean value for BMI is 31
```

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

```
+-----------+-------+-------------+-------------+-------+----+------------------------
-+---+-------+
|Pregnancies|Glucose|BloodPressure|SkinThickness|Insulin| BMI|DiabetesPedigreeFunctio
n|Age|Outcome|
+-----------+-------+-------------+-------------+-------+----+------------------------
-+---+-------+
|          6|    148|           72|           35|     79|33.6|                    0.62
7| 50|      1|
|          1|     85|           66|           29|     79|26.6|                    0.35
1| 31|      0|
|          8|    183|           64|           20|     79|23.3|                    0.67
2| 32|      1|
|          1|     89|           66|           23|     94|28.1|                    0.16
7| 21|      0|
|          0|    137|           40|           35|    168|43.1|                    2.28
8| 33|      1|
|          5|    116|           74|           20|     79|25.6|                    0.20
1| 30|      0|
|          3|     78|           50|           32|     88|31.0|                    0.24
8| 26|      1|
|         10|    115|           69|           20|     79|35.3|                    0.13
4| 29|      0|
|          2|    197|           70|           45|    543|30.5|                    0.15
```

```
8| 53|      1|
|           8|    125|           96|           20|     79|31.0|                    0.23
2| 54|      1|
|           4|    110|           92|           20|     79|37.6|                    0.19
1| 30|      0|
|          10|    168|           74|           20|     79|38.0|                    0.53
7| 34|      1|
|          10|    139|           80|           20|     79|27.1|                    1.44
1| 57|      0|
|           1|    189|           60|           23|    846|30.1|                    0.39
8| 59|      1|
|           5|    166|           72|           19|    175|25.8|                    0.58
7| 51|      1|
|           7|    100|           69|           20|     79|30.0|                    0.48
4| 32|      1|
|           0|    118|           84|           47|    230|45.8|                    0.55
1| 31|      1|
|           7|    107|           74|           20|     79|29.6|                    0.25
4| 31|      1|
|           1|    103|           30|           38|     83|43.3|                    0.18
3| 33|      0|
|           1|    115|           70|           30|     96|34.6|                    0.52
9| 32|      1|
+------------+-------+-------------+-------------+-------+----+-----------------------
-+---+-------+
only showing top 20 rows
```

Correlation Analysis & Feature Selection

In [ ]:
```
for col in df.columns[:8]:
  print(f'Correlation to target for {col} feature is {df.stat.corr("Outcome", col)}')
```

```
Correlation to target for Pregnancies feature is 0.22189815303398638
Correlation to target for Glucose feature is 0.49288410274882094
Correlation to target for BloodPressure feature is 0.16287909949861834
Correlation to target for SkinThickness feature is 0.1718568141765 64
Correlation to target for Insulin feature is 0.17869558803050842
Correlation to target for BMI feature is 0.31289043493401536
Correlation to target for DiabetesPedigreeFunction feature is 0.17384406565296007
Correlation to target for Age feature is 0.23835598302719757
```

In [ ]:
```
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=['Pregnancies','Glucose','BloodPressure',
                                       'SkinThickness','Insulin','BMI',
                                       'DiabetesPedigreeFunction','Age'],
                            outputCol='features')
output_data = assembler.transform(df)
```

In [ ]:
```
output_data.printSchema()
```

```
root
 |-- Pregnancies: integer (nullable = true)
 |-- Glucose: integer (nullable = true)
 |-- BloodPressure: integer (nullable = true)
 |-- SkinThickness: integer (nullable = true)
 |-- Insulin: integer (nullable = true)
 |-- BMI: double (nullable = true)
 |-- DiabetesPedigreeFunction: double (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Outcome: integer (nullable = true)
 |-- features: vector (nullable = true)
```

In [ ]:
```
output_data.show()
```

```
+------------+-------+-------------+-------------+-------+----+-----------------------
-+---+-------+--------------------+
```

```
|Pregnancies|Glucose|BloodPressure|SkinThickness|Insulin| BMI|DiabetesPedigreeFunctio
n|Age|Outcome|            features|
+-----------+-------+-------------+-------------+-------+----+----------------------
-+---+-------+--------------------+
|          6|    148|           72|           35|     79|33.6|                   0.62
7| 50|      1|[6.0,148.0,72.0,3...|
|          1|     85|           66|           29|     79|26.6|                   0.35
1| 31|      0|[1.0,85.0,66.0,29...|
|          8|    183|           64|           20|     79|23.3|                   0.67
2| 32|      1|[8.0,183.0,64.0,2...|
|          1|     89|           66|           23|     94|28.1|                   0.16
7| 21|      0|[1.0,89.0,66.0,23...|
|          0|    137|           40|           35|    168|43.1|                   2.28
8| 33|      1|[0.0,137.0,40.0,3...|
|          5|    116|           74|           20|     79|25.6|                   0.20
1| 30|      0|[5.0,116.0,74.0,2...|
|          3|     78|           50|           32|     88|31.0|                   0.24
8| 26|      1|[3.0,78.0,50.0,32...|
|         10|    115|           69|           20|     79|35.3|                   0.13
4| 29|      0|[10.0,115.0,69.0,...|
|          2|    197|           70|           45|    543|30.5|                   0.15
8| 53|      1|[2.0,197.0,70.0,4...|
|          8|    125|           96|           20|     79|31.0|                   0.23
2| 54|      1|[8.0,125.0,96.0,2...|
|          4|    110|           92|           20|     79|37.6|                   0.19
1| 30|      0|[4.0,110.0,92.0,2...|
|         10|    168|           74|           20|     79|38.0|                   0.53
7| 34|      1|[10.0,168.0,74.0,...|
|         10|    139|           80|           20|     79|27.1|                   1.44
1| 57|      0|[10.0,139.0,80.0,...|
|          1|    189|           60|           23|    846|30.1|                   0.39
8| 59|      1|[1.0,189.0,60.0,2...|
|          5|    166|           72|           19|    175|25.8|                   0.58
7| 51|      1|[5.0,166.0,72.0,1...|
|          7|    100|           69|           20|     79|30.0|                   0.48
4| 32|      1|[7.0,100.0,69.0,2...|
|          0|    118|           84|           47|    230|45.8|                   0.55
1| 31|      1|[0.0,118.0,84.0,4...|
|          7|    107|           74|           20|     79|29.6|                   0.25
4| 31|      1|[7.0,107.0,74.0,2...|
|          1|    103|           30|           38|     83|43.3|                   0.18
3| 33|      0|[1.0,103.0,30.0,3...|
|          1|    115|           70|           30|     96|34.6|                   0.52
9| 32|      1|[1.0,115.0,70.0,3...|
+-----------+-------+-------------+-------------+-------+----+----------------------
-+---+-------+--------------------+
only showing top 20 rows
```

Build the Model

In [ ]:
```
from pyspark.ml.classification import LogisticRegression

final_data = output_data.select(['features','Outcome'])
```

In [ ]:
```
final_data.printSchema()
```

```
root
 |-- features: vector (nullable = true)
 |-- Outcome: integer (nullable = true)
```

In [ ]:
```
train, test = final_data.randomSplit([0.7,0.3])
models = LogisticRegression(labelCol='Outcome')
model = models.fit(train)
```

In [ ]:
```
summary = model.summary
summary.predictions.describe().show()
```

```
+-------+-------------------+-------------------+
|summary|            Outcome|         prediction|
+-------+-------------------+-------------------+
|  count|                529|                529|
|   mean|0.34782608695652173|0.2684310018903592|
| stddev|0.47673129462279645| 0.443562535587099|
|    min|                0.0|                0.0|
|    max|                1.0|                1.0|
+-------+-------------------+-------------------+
```

In [ ]:
```
from pyspark.ml.evaluation import BinaryClassificationEvaluator

predictions = model.evaluate(test)
```

In [ ]:
```
predictions.predictions.show(10)
```

```
+--------------------+-------+--------------------+--------------------+----------+
|            features|Outcome|       rawPrediction|         probability|prediction|
+--------------------+-------+--------------------+--------------------+----------+
|[0.0,57.0,60.0,20...|      0|[3.04252988717473...|[0.95445892256214...|       0.0|
|[0.0,94.0,69.0,20...|      0|[2.88041125013767...|[0.94686955641171...|       0.0|
|[0.0,95.0,80.0,45...|      0|[2.33089743991755...|[0.91140382898712...|       0.0|
|[0.0,99.0,69.0,20...|      0|[3.24039807390352...|[0.96232654398539...|       0.0|
|[0.0,100.0,88.0,6...|      0|[0.42333897573322...|[0.60428196071239...|       0.0|
|[0.0,102.0,75.0,2...|      0|[2.34434110250901...|[0.91248337593531...|       0.0|
|[0.0,102.0,78.0,4...|      0|[2.38658692297128...|[0.91579875419639...|       0.0|
|[0.0,105.0,90.0,2...|      0|[2.34476162879304...|[0.91251695227349...|       0.0|
|[0.0,106.0,70.0,3...|      0|[1.41510938421794...|[0.80457057706697...|       0.0|
|[0.0,107.0,60.0,2...|      0|[2.77537580901381...|[0.94133058337379...|       0.0|
+--------------------+-------+--------------------+--------------------+----------+
only showing top 10 rows
```

In [ ]:
```
evaluator = BinaryClassificationEvaluator(rawPredictionCol = 'rawPrediction', labelCol
evaluator.evaluate(model.transform(test))
```

Out[ ]:
```
0.8410138248847925
```

In [ ]:
```
import os

# Define the model save path
save_path = 'model'

# Check if the directory exists, if not create it
if not os.path.exists(save_path):
    os.makedirs(save_path)

# Save the model with overwrite option
try:
    model.write().overwrite().save('model')
except Exception as e:
    print(f"Error occurred while saving the model: {e}")
```

In [ ]:
```
from pyspark.ml.classification import LogisticRegressionModel

model2 = LogisticRegressionModel.load('model')
```

In [ ]:
```
df_test = spark.read.csv('new_test.csv', header=True, inferSchema=True)
```

In [ ]:
```
df_test.printSchema()
```

```
root
 |-- Pregnancies: integer (nullable = true)
```

```
 |-- Glucose: integer (nullable = true)
 |-- BloodPressure: integer (nullable = true)
 |-- SkinThickness: integer (nullable = true)
 |-- Insulin: integer (nullable = true)
 |-- BMI: double (nullable = true)
 |-- DiabetesPedigreeFunction: double (nullable = true)
 |-- Age: integer (nullable = true)
```

In [ ]:
```
test_data = assembler.transform(df_test)
```

In [ ]:
```
test_data.printSchema()
```

```
root
 |-- Pregnancies: integer (nullable = true)
 |-- Glucose: integer (nullable = true)
 |-- BloodPressure: integer (nullable = true)
 |-- SkinThickness: integer (nullable = true)
 |-- Insulin: integer (nullable = true)
 |-- BMI: double (nullable = true)
 |-- DiabetesPedigreeFunction: double (nullable = true)
 |-- Age: integer (nullable = true)
 |-- features: vector (nullable = true)
```

In [ ]:
```
results = model2.transform(test_data)
results.printSchema()
```

```
root
 |-- Pregnancies: integer (nullable = true)
 |-- Glucose: integer (nullable = true)
 |-- BloodPressure: integer (nullable = true)
 |-- SkinThickness: integer (nullable = true)
 |-- Insulin: integer (nullable = true)
 |-- BMI: double (nullable = true)
 |-- DiabetesPedigreeFunction: double (nullable = true)
 |-- Age: integer (nullable = true)
 |-- features: vector (nullable = true)
 |-- rawPrediction: vector (nullable = true)
 |-- probability: vector (nullable = true)
 |-- prediction: double (nullable = false)
```

In [ ]:
```
results.select(['features','prediction']).show()
```

```
+--------------------+----------+
|            features|prediction|
+--------------------+----------+
|[1.0,190.0,78.0,3...|       1.0|
|[0.0,80.0,84.0,36...|       0.0|
|[2.0,138.0,82.0,4...|       1.0|
|[1.0,110.0,63.0,4...|       1.0|
+--------------------+----------+
```