```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, sum
from pyspark.ml.feature import StringIndexer,VectorAssembler,OneHotEncoder
from pyspark.ml import Pipeline
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator


spark = SparkSession.builder.appName("EDA").getOrCreate()
df = spark.read.csv("AlzDataset.csv",header=True,inferSchema=True)


#EDA
print("Displaying first 5 rows of dataset : ")
df.show(5)

print("Displaying schema of dataset : ")
df.printSchema()

print("Displaying summary statistics of dataset : ")
df.describe().show()

print(f"Total Rows: {df.count()}")
print(f"Total Columns: {len(df.columns)}")
```

```
Displaying first 5 rows of dataset :
+------------+---+------+---------------+----+----------------------+--------------+------------------+--------+------------+----
|     Country|Age|Gender|Education Level| BMI|Physical Activity Level|Smoking Status|Alcohol Consumption|Diabetes|Hypertension|Chole
+------------+---+------+---------------+----+----------------------+--------------+------------------+--------+------------+----
|       Spain| 90|  Male|              1|33.0|                Medium|         Never|       Occasionally|      No|          No|
|   Argentina| 72|  Male|              7|29.9|                Medium|        Former|             Never|      No|          No|
|South Africa| 86|Female|             19|22.9|                  High|       Current|       Occasionally|      No|         Yes|
|       China| 53|  Male|             17|31.2|                   Low|         Never|          Regularly|     Yes|          No|
|      Sweden| 58|Female|              3|30.0|                  High|        Former|             Never|     Yes|          No|
+------------+---+------+---------------+----+----------------------+--------------+------------------+--------+------------+----
only showing top 5 rows

Displaying schema of dataset :
root
 |-- Country: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Education Level: integer (nullable = true)
 |-- BMI: double (nullable = true)
 |-- Physical Activity Level: string (nullable = true)
 |-- Smoking Status: string (nullable = true)
 |-- Alcohol Consumption: string (nullable = true)
 |-- Diabetes: string (nullable = true)
 |-- Hypertension: string (nullable = true)
 |-- Cholesterol Level: string (nullable = true)
 |-- Family History of Alzheimer’s: string (nullable = true)
 |-- Cognitive Test Score: integer (nullable = true)
 |-- Depression Level: string (nullable = true)
 |-- Sleep Quality: string (nullable = true)
 |-- Dietary Habits: string (nullable = true)
 |-- Air Pollution Exposure: string (nullable = true)
 |-- Employment Status: string (nullable = true)
 |-- Marital Status: string (nullable = true)
 |-- Genetic Risk Factor (APOE-ε4 allele): string (nullable = true)
 |-- Social Engagement Level: string (nullable = true)
 |-- Income Level: string (nullable = true)
 |-- Stress Levels: string (nullable = true)
 |-- Urban vs Rural Living: string (nullable = true)
 |-- Alzheimer_Diagnosis: string (nullable = true)

Displaying summary statistics of dataset :
+-------+---------+------------------+------+-----------------+-----------------+-----------------------+--------------+-----------
|summary|  Country|               Age|Gender|  Education Level|              BMI|Physical Activity Level|Smoking Status|Alcohol Consu
+-------+---------+------------------+------+-----------------+-----------------+-----------------------+--------------+-----------
|  count|    74283|             74283| 74283|            74283|            74283|                  74283|         74283|
|   mean|     NULL|71.96470255643956|  NULL|9.487513966856481|26.78063890796039|                   NULL|          NULL|
| stddev|     NULL| 12.9807479595137|  NULL|5.757020114227965| 4.76467948324257|                   NULL|          NULL|
|    min|Argentina|                50|Female|                0|             18.5|                   High|       Current|
|    max|      USA|                94|  Male|               19|             35.0|                 Medium|         Never|          Reg
+-------+---------+------------------+------+-----------------+-----------------+-----------------------+--------------+-----------

Total Rows: 74283
Total Columns: 25
```

```
#Checking for null values
print("Displaying number of Null values in each column : ")
df.select([sum(col(c).isNull().cast("int")).alias(c) for c in df.columns]).show()
```

```
Displaying number of Null values in each column :
+-------+---+------+---------------+---+----------------------+-------------+------------------+--------+------------+----------
|Country|Age|Gender|Education Level|BMI|Physical Activity Level|Smoking Status|Alcohol Consumption|Diabetes|Hypertension|Cholesterol
+-------+---+------+---------------+---+----------------------+-------------+------------------+--------+------------+----------
|      0|  0|     0|              0|  0|                     0|            0|                 0|       0|           0|
+-------+---+------+---------------+---+----------------------+-------------+------------------+--------+------------+----------
```

```
#Pre-processing
categorical_columns = [item[0] for item in df.dtypes if item[1].startswith('string')]
print(f"No of Categorical columns : {len(categorical_columns)}")
print(categorical_columns)
```

```
No of Categorical columns : 21
['Country', 'Gender', 'Physical Activity Level', 'Smoking Status', 'Alcohol Consumption', 'Diabetes', 'Hypertension', 'Cholesterol l
```

```
numerical_columns = [item[0] for item in df.dtypes if item[1].startswith(('double','int'))]
print(f"No of Numerical columns : {len(numerical_columns)}")
print(numerical_columns)
```

```
No of Numerical columns : 4
['Age', 'Education Level', 'BMI', 'Cognitive Test Score']
```

```
#Target Column
target_col = "Alzheimer_Diagnosis"
```

```
#Encoding
print("Encoding Categorical features")
indexers = [StringIndexer(inputCol=col, outputCol=col+"_index", handleInvalid="keep") for col in categorical_columns if col != target_c
print(indexers)
```

```
Encoding Categorical features
[StringIndexer_2c4f13019ea8, StringIndexer_4c5686af498a, StringIndexer_d8abe661fb02, StringIndexer_8fc009aa38e9, StringIndexer_fc541
```

```
# Encode target column separately
target_indexer = StringIndexer(inputCol=target_col, outputCol="label", handleInvalid="keep")
```

```
# Feature assembler (excluding target)
feature_columns = [col+"_index" for col in categorical_columns if col != target_col] + numerical_columns
assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")
```

```
# Create pipeline (encoding + feature assembling)
pipeline = Pipeline(stages=indexers + [target_indexer, assembler])
```

```
# Transform dataset
df = pipeline.fit(df).transform(df)
```

```
# Splitting data (80% train, 20% test)
train_df, test_df = df.randomSplit([0.8, 0.2], seed=42)
```

```
# Train Decision Tree Classifier
dt = DecisionTreeClassifier(labelCol="label", featuresCol="features")
model = dt.fit(train_df)
```

```
# Predictions
predictions = model.transform(test_df)
print(predictions.show())
```

```
+---------+---+------+---------------+----+----------------------+-------------+------------------+--------+------------+---------
|  Country|Age|Gender|Education Level| BMI|Physical Activity Level|Smoking Status|Alcohol Consumption|Diabetes|Hypertension|Cholester
+---------+---+------+---------------+----+----------------------+-------------+------------------+--------+------------+---------
|Argentina| 50|Female|              2|32.4|                  High|        Never|             Never|      No|          No|
|Argentina| 50|Female|              6|25.8|                   Low|       Former|      Occasionally|     Yes|         Yes|
|Argentina| 50|Female|              6|29.7|                   Low|      Current|         Regularly|      No|          No|
|Argentina| 50|Female|              7|33.8|                   Low|      Current|             Never|     Yes|         Yes|
|Argentina| 50|Female|             12|25.4|                  High|      Current|             Never|     Yes|         Yes|
|Argentina| 50|Female|             14|23.0|                   Low|       Former|      Occasionally|      No|         Yes|
|Argentina| 50|  Male|              1|27.0|                  High|       Former|      Occasionally|      No|          No|
|Argentina| 50|  Male|              3|19.1|                Medium|       Former|             Never|      No|          No|
|Argentina| 50|  Male|              9|24.7|                  High|       Former|             Never|      No|         Yes|
|Argentina| 50|  Male|              9|33.6|                Medium|      Current|             Never|      No|          No|
|Argentina| 50|  Male|              9|33.8|                   Low|       Former|         Regularly|      No|          No|
|Argentina| 50|  Male|             10|30.3|                   Low|      Current|         Regularly|      No|          No|
|Argentina| 50|  Male|             11|20.9|                  High|       Former|         Regularly|      No|          No|
```

```
|Argentina| 50|  Male|             13|25.7|                  High|        Former|             Never|      No|          No|
|Argentina| 50|  Male|             16|32.8|                Medium|       Current|       Occasionally|      No|          No|
|Argentina| 50|  Male|             18|30.9|                   Low|         Never|             Never|      No|         Yes|
|Argentina| 51|Female|              1|25.8|                   Low|        Former|             Never|      No|          No|
|Argentina| 51|Female|             15|28.7|                Medium|         Never|       Occasionally|      No|          No|
|Argentina| 51|  Male|              0|20.2|                   Low|         Never|             Never|      No|          No|
|Argentina| 51|  Male|              2|28.7|                   Low|       Current|       Occasionally|     Yes|          No|
+---------+---+------+---------------+----+----------------------+--------------+------------------+--------+------------+---------
only showing top 20 rows

None
```

```
last_10 = predictions.tail(10)
for row in last_10:
    print(row)
```

```
2, 50.0]), rawPrediction=DenseVector([4607.0, 5287.0, 0.0]), probability=DenseVector([0.4656, 0.5344, 0.0]), prediction=1.0)
 32.7, 23: 91.0}), rawPrediction=DenseVector([4607.0, 5287.0, 0.0]), probability=DenseVector([0.4656, 0.5344, 0.0]), prediction=1.0)
8, 78.0]), rawPrediction=DenseVector([1794.0, 4491.0, 0.0]), probability=DenseVector([0.2854, 0.7146, 0.0]), prediction=1.0)
26.1, 36.0]), rawPrediction=DenseVector([4607.0, 5287.0, 0.0]), probability=DenseVector([0.4656, 0.5344, 0.0]), prediction=1.0)
wPrediction=DenseVector([1794.0, 4491.0, 0.0]), probability=DenseVector([0.2854, 0.7146, 0.0]), prediction=1.0)
awPrediction=DenseVector([1794.0, 4491.0, 0.0]), probability=DenseVector([0.2854, 0.7146, 0.0]), prediction=1.0)
, rawPrediction=DenseVector([895.0, 4328.0, 0.0]), probability=DenseVector([0.1714, 0.8286, 0.0]), prediction=1.0)
.0]), rawPrediction=DenseVector([4607.0, 5287.0, 0.0]), probability=DenseVector([0.4656, 0.5344, 0.0]), prediction=1.0)
]), rawPrediction=DenseVector([1794.0, 4491.0, 0.0]), probability=DenseVector([0.2854, 0.7146, 0.0]), prediction=1.0)
Prediction=DenseVector([4607.0, 5287.0, 0.0]), probability=DenseVector([0.4656, 0.5344, 0.0]), prediction=1.0)
```

```
# Evaluate Model
evaluator = MulticlassClassificationEvaluator(labelCol="label", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)

print(f"Test Accuracy: {accuracy}")
```

```
Test Accuracy: 0.7244891077089094
```