VASANTDADA PATIL PRATISHTHAN'S COLLEGE OF ENGINEERING AND VISUAL ARTS

ISO 9001:2015 Certified Institute

Department of Information Technology

NBA Accredited Course (Dated 01/07/2024 to 30/06/2027)

EXPERIMENT – 9

Aim: Exploratory data analysis using Apache Spark and Pandas.

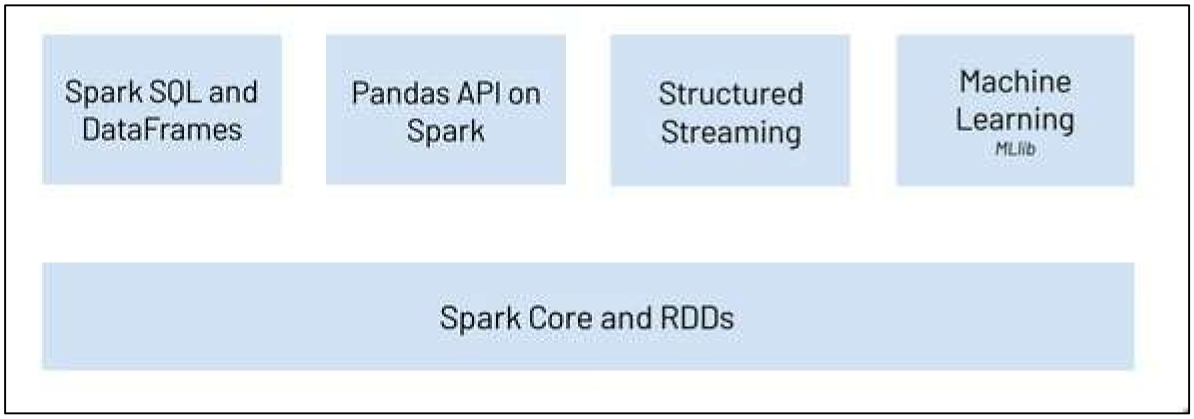
Theory:

Apache Spark

PySpark is the Python API for Apache Spark. It enables you to perform real-time, large-scale data processing in a distributed environment using Python. It also provides a PySpark shell for interactively analyzing your data.

PySpark combines Python's learnability and ease of use with the power of Apache Spark to enable processing and analysis of data at any size for everyone familiar with Python.

PySpark supports all of Spark's features such as Spark SQL, DataFrames, Structured Streaming, Machine Learning (MLlib) and Spark Core.

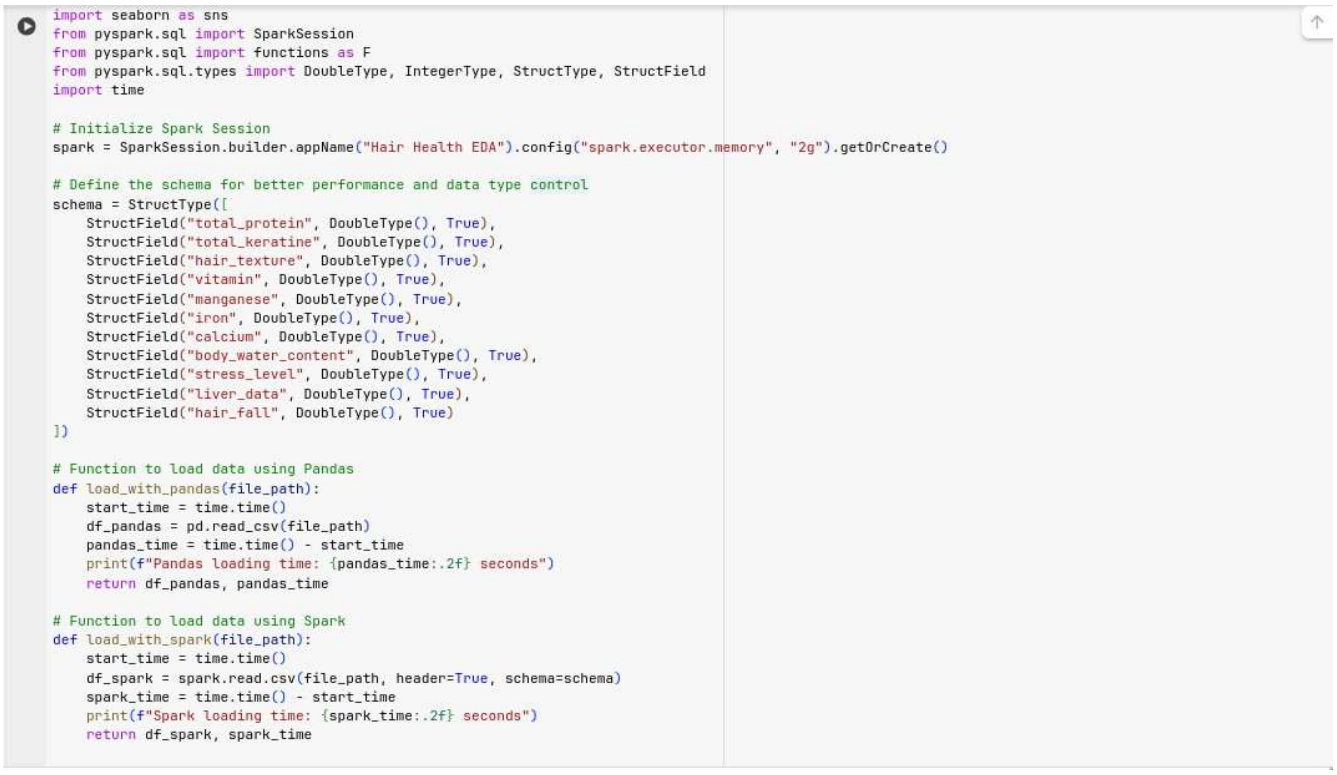


Spark SQL and DataFrames

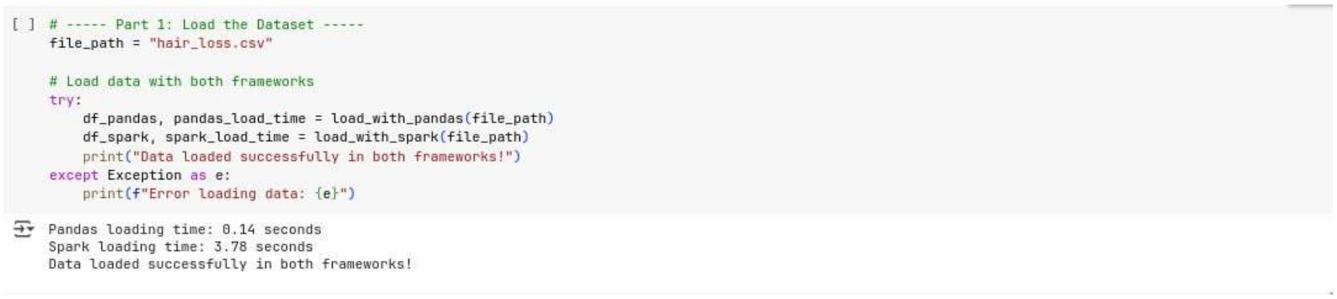
Spark SQL is Apache Spark's module for working with structured data. It allows you to seamlessly mix SQL queries with Spark programs. With PySpark DataFrames you can efficiently read, write, transform, and analyze data using Python and SQL. Whether you use Python or SQL, the same underlying execution engine is used so you will always leverage the full power of Spark.

```
> pip install pyspark
```

Program:

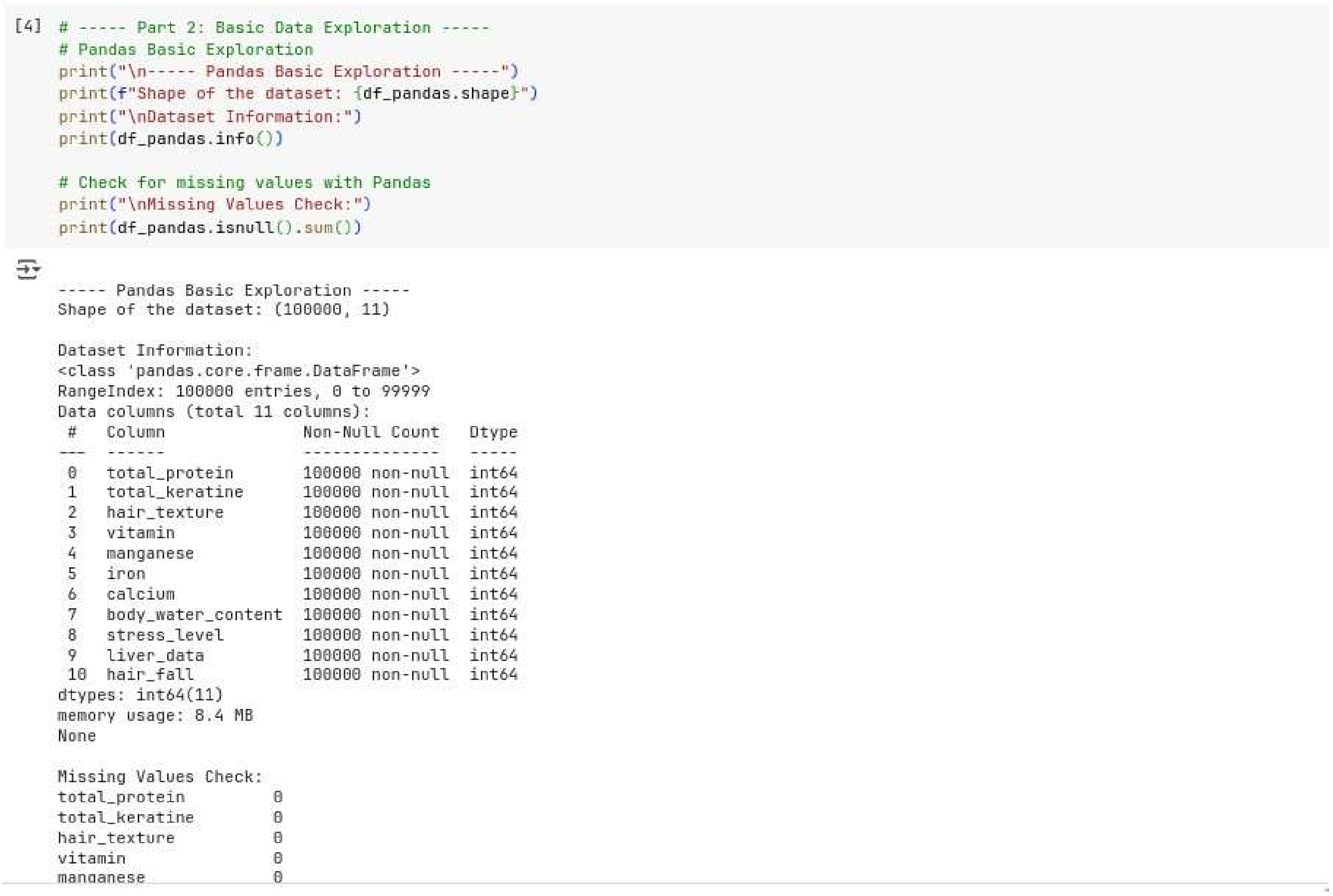


```
import seaborn as sns
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql.types import DoubleType, IntegerType, StructType, StructField
import time

# Initialize Spark Session
spark = SparkSession.builder.appName("Hair Health EDA").config("spark.executor.memory", "2g").getOrCreate()

# Define the schema for better performance and data type control
schema = StructType([
    StructField("total_protein", DoubleType(), True),
    StructField("total_keratine", DoubleType(), True),
    StructField("hair_texture", DoubleType(), True),
    StructField("vitamin", DoubleType(), True),
    StructField("manganese", DoubleType(), True),
    StructField("iron", DoubleType(), True),
    StructField("calcium", DoubleType(), True),
    StructField("body_water_content", DoubleType(), True),
    StructField("stress_level", DoubleType(), True),
    StructField("liver_data", DoubleType(), True),
    StructField("hair_fall", DoubleType(), True)
])

# Function to load data using Pandas
def load_with_pandas(file_path):
    start_time = time.time()
    df_pandas = pd.read_csv(file_path)
    pandas_time = time.time() - start_time
    print(f"Pandas loading time: {pandas_time:.2f} seconds")
    return df_pandas, pandas_time

# Function to load data using Spark
def load_with_spark(file_path):
    start_time = time.time()
    df_spark = spark.read.csv(file_path, header=True, schema=schema)
    spark_time = time.time() - start_time
    print(f"Spark loading time: {spark_time:.2f} seconds")
    return df_spark, spark_time
```



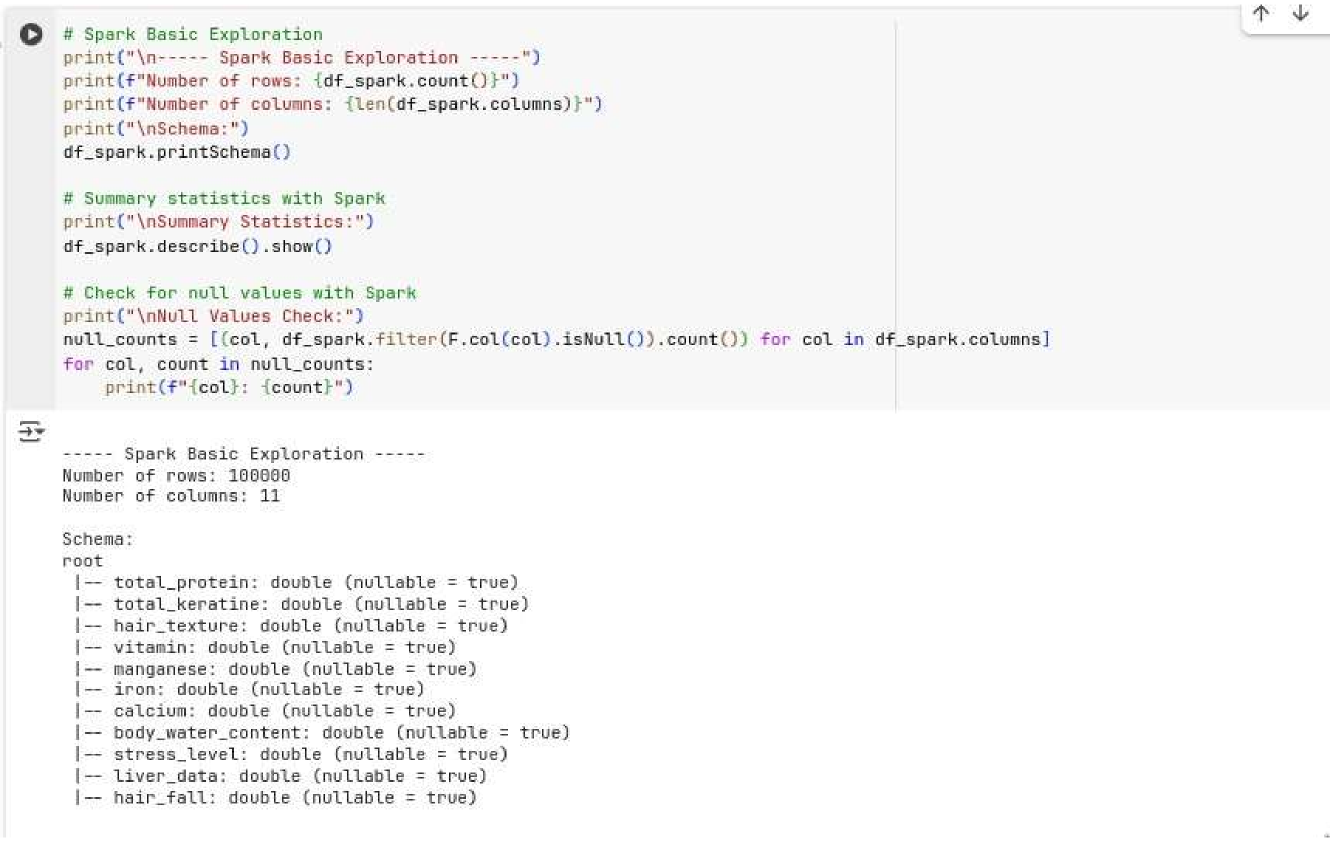
```
[ ] # ----- Part 1: Load the Dataset -----
    file_path = "hair_loss.csv"

    # Load data with both frameworks
    try:
        df_pandas, pandas_load_time = load_with_pandas(file_path)
        df_spark, spark_load_time = load_with_spark(file_path)
        print("Data loaded successfully in both frameworks!")
    except Exception as e:
        print(f"Error loading data: {e}")
```

```
Pandas loading time: 0.14 seconds
Spark loading time: 3.78 seconds
Data loaded successfully in both frameworks!
```



```
[4] # ----- Part 2: Basic Data Exploration -----
    # Pandas Basic Exploration
    print("\n----- Pandas Basic Exploration -----")
    print(f"Shape of the dataset: {df_pandas.shape}")
    print("\nDataset Information:")
    print(df_pandas.info())

    # Check for missing values with Pandas
    print("\nMissing Values Check:")
    print(df_pandas.isnull().sum())
```

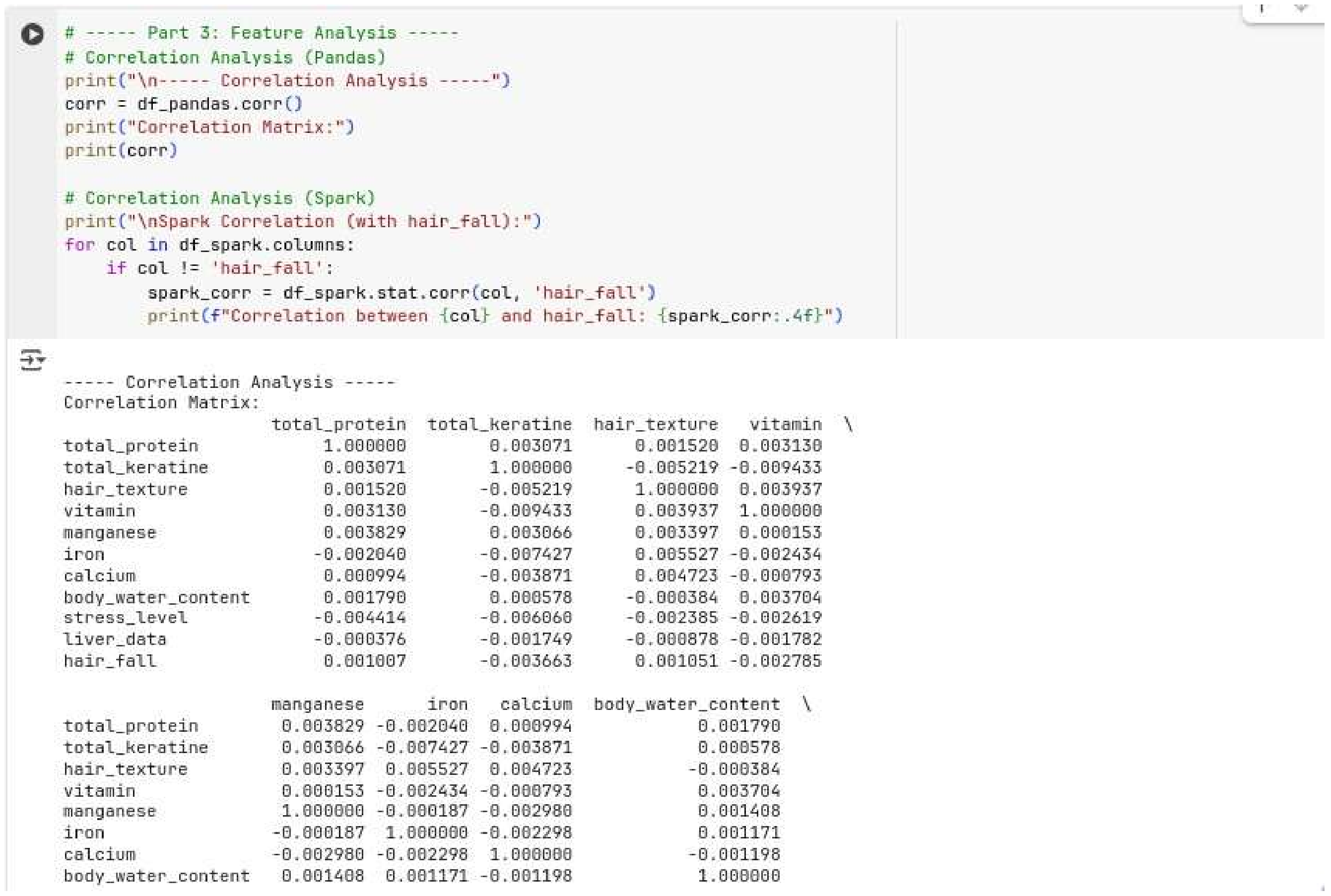
```
----- Pandas Basic Exploration -----
Shape of the dataset: (100000, 11)

Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 11 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   total_protein       100000 non-null  int64
 1   total_keratine      100000 non-null  int64
 2   hair_texture        100000 non-null  int64
 3   vitamin             100000 non-null  int64
 4   manganese           100000 non-null  int64
 5   iron                100000 non-null  int64
 6   calcium             100000 non-null  int64
 7   body_water_content  100000 non-null  int64
 8   stress_level        100000 non-null  int64
 9   liver_data          100000 non-null  int64
 10  hair_fall           100000 non-null  int64
dtypes: int64(11)
memory usage: 8.4 MB
None

Missing Values Check:
total_protein         0
total_keratine        0
hair_texture          0
vitamin               0
manganese             0
```



```python
# Spark Basic Exploration
print("\n----- Spark Basic Exploration -----")
print(f"Number of rows: {df_spark.count()}")
print(f"Number of columns: {len(df_spark.columns)}")
print("\nSchema:")
df_spark.printSchema()

# Summary statistics with Spark
print("\nSummary Statistics:")
df_spark.describe().show()

# Check for null values with Spark
print("\nNull Values Check:")
null_counts = [(col, df_spark.filter(F.col(col).isNull()).count()) for col in df_spark.columns]
for col, count in null_counts:
    print(f"{col}: {count}")
```

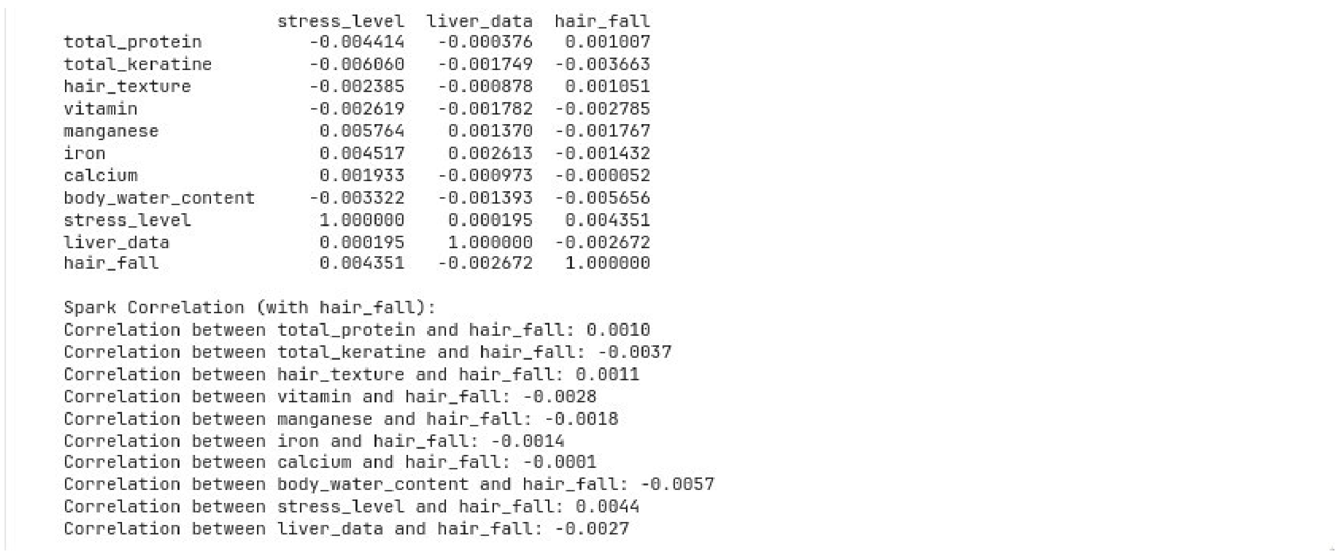
```
----- Spark Basic Exploration -----
Number of rows: 100000
Number of columns: 11

Schema:
root
 |-- total_protein: double (nullable = true)
 |-- total_keratine: double (nullable = true)
 |-- hair_texture: double (nullable = true)
 |-- vitamin: double (nullable = true)
 |-- manganese: double (nullable = true)
 |-- iron: double (nullable = true)
 |-- calcium: double (nullable = true)
 |-- body_water_content: double (nullable = true)
 |-- stress_level: double (nullable = true)
 |-- liver_data: double (nullable = true)
 |-- hair_fall: double (nullable = true)
```

```
Summary Statistics:
+-------+-----------------+-----------------+------------------+------------------+------------------+------------------+
|summary|    total_protein|   total_keratine|      hair_texture|           vitamin|         manganese|              iron|
+-------+-----------------+-----------------+------------------+------------------+------------------+------------------+
|  count|           100000|           100000|            100000|            100000|            100000|            100000|
|   mean|        249.60834|         248.9176|          49.57226|         249.94973|         249.55848|         249.09926|
| stddev|144.69885092846712|144.8711280644889|29.227406674308472|144.24063940702558|144.28359455477286|144.34127348573935|
|    min|              0.0|              0.0|               0.0|               0.0|               0.0|               0.0|
|    max|           2999.0|           4681.0|            1400.0|             499.0|             499.0|             499.0|
+-------+-----------------+-----------------+------------------+------------------+------------------+------------------+


Null Values Check:
total_protein: 0
total_keratine: 0
hair_texture: 0
vitamin: 0
manganese: 0
iron: 0
calcium: 0
body_water_content: 0
stress_level: 0
liver_data: 0
hair_fall: 0
```



```python
# ----- Part 3: Feature Analysis -----
# Correlation Analysis (Pandas)
print("\n----- Correlation Analysis -----")
corr = df_pandas.corr()
print("Correlation Matrix:")
print(corr)

# Correlation Analysis (Spark)
print("\nSpark Correlation (with hair_fall):")
for col in df_spark.columns:
    if col != 'hair_fall':
        spark_corr = df_spark.stat.corr(col, 'hair_fall')
        print(f"Correlation between {col} and hair_fall: {spark_corr:.4f}")
```

```
----- Correlation Analysis -----
Correlation Matrix:
                   total_protein  total_keratine  hair_texture   vitamin  \
total_protein           1.000000        0.003071      0.001520  0.003130
total_keratine          0.003071        1.000000     -0.005219 -0.009433
hair_texture            0.001520       -0.005219      1.000000  0.003937
vitamin                 0.003130       -0.009433      0.003937  1.000000
manganese               0.003829        0.003066      0.003397  0.000153
iron                   -0.002040       -0.007427      0.005527 -0.002434
calcium                 0.000994       -0.003871      0.004723 -0.000793
body_water_content      0.001790        0.000578     -0.000384  0.003704
stress_level           -0.004414       -0.006060     -0.002385 -0.002619
liver_data             -0.000376       -0.001749     -0.000878 -0.001782
hair_fall               0.001007       -0.003663      0.001051 -0.002785

                   manganese      iron   calcium  body_water_content  \
total_protein       0.003829 -0.002040  0.000994            0.001790
total_keratine      0.003066 -0.007427 -0.003871            0.000578
hair_texture        0.003397  0.005527  0.004723           -0.000384
vitamin             0.000153 -0.002434 -0.000793            0.003704
manganese           1.000000 -0.000187 -0.002980            0.001408
iron               -0.000187  1.000000 -0.002298            0.001171
calcium            -0.002980 -0.002298  1.000000           -0.001198
body_water_content  0.001408  0.001171 -0.001198            1.000000
```



```
                    stress_level  liver_data  hair_fall
total_protein          -0.004414   -0.000376   0.001007
total_keratine         -0.006060   -0.001749  -0.003663
hair_texture           -0.002385   -0.000878   0.001051
vitamin                -0.002619   -0.001782  -0.002785
manganese               0.005764    0.001370  -0.001767
iron                    0.004517    0.002613  -0.001432
calcium                 0.001933   -0.000973  -0.000052
body_water_content     -0.003322   -0.001393  -0.005656
stress_level            1.000000    0.000195   0.004351
liver_data              0.000195    1.000000  -0.002672
hair_fall               0.004351   -0.002672   1.000000

Spark Correlation (with hair_fall):
Correlation between total_protein and hair_fall: 0.0010
Correlation between total_keratine and hair_fall: -0.0037
Correlation between hair_texture and hair_fall: 0.0011
Correlation between vitamin and hair_fall: -0.0028
Correlation between manganese and hair_fall: -0.0018
Correlation between iron and hair_fall: -0.0014
Correlation between calcium and hair_fall: -0.0001
Correlation between body_water_content and hair_fall: -0.0057
Correlation between stress_level and hair_fall: 0.0044
Correlation between liver_data and hair_fall: -0.0027
```

Conclusion: Thus, we have successfully implemented Exploratory data analysis using Apache Spark and Pandas