```
dbutils.fs.mkdirs("/FileStore/tables")
```

```
True
```

```
display(dbutils.fs.ls("/FileStore/shared_uploads/
azuser3611_mml.local@techademy.com/"))
```

```
#Extract
weather_df = spark.read.option("header",
True).csv("/FileStore/shared_uploads/azuser3611_mml.local@techademy.co
m/weather_data.csv", inferSchema=True)
soil_df = spark.read.option("header",
True).csv("/FileStore/shared_uploads/azuser3611_mml.local@techademy.co
m/soil_data.csv", inferSchema=True)
fertilizer_df = spark.read.option("header",
True).csv("/FileStore/shared_uploads/azuser3611_mml.local@techademy.co
m/fertilizer_usage.csv", inferSchema=True)
```

```
#Transform - Aggregate weather
from pyspark.sql.functions import avg, sum

weather_agg = weather_df.groupBy("farm_id").agg(
    avg("average_temperature").alias("avg_temperature"),
    sum("total_rainfall").alias("tot_rainfall")
)
```

```
weather_df.printSchema()
```

```
root
 |-- farm_id: integer (nullable = true)
 |-- average_temperature: double (nullable = true)
 |-- total_rainfall: double (nullable = true)
```

```
#Join data
joined_df = fertilizer_df.join(soil_df, "farm_id", "inner") \
                         .join(weather_agg, "farm_id", "inner")
```

```
#Handling Missing Values
from pyspark.sql.functions import mean

for col in ["soil_quality_index", "fertilizer_used",
"average_temperature", "tot_rainfall"]:
    mean_val = joined_df.select(mean(col)).first()[0]
    joined_df = joined_df.na.fill({col: mean_val})
```

```
# Normalize Numerical Columns
from pyspark.ml.feature import VectorAssembler, StandardScaler

assembler = VectorAssembler(
    inputCols=["soil_quality_index", "fertilizer_used",
"avg_temperature", "tot_rainfall"],
    outputCol="features"
)
assembled_df = assembler.transform(joined_df)

scaler = StandardScaler(
    inputCol="features",
    outputCol="scaled_features",
    withStd=True,
    withMean=True
)
scaler_model = scaler.fit(assembled_df)
scaled_df = scaler_model.transform(assembled_df)

display(scaled_df.select("farm_id", "scaled_features"))

joined_df.write.option("header",
"true").mode("overwrite").csv("/FileStore/tables/farm_yield_cleaned_da
ta.csv")

display(dbutils.fs.ls("/FileStore/tables/
farm_yield_cleaned_data.csv"))
```