BDA PROJECT

BIG DATA ANALYTICS MINI PROJECT

T.SAI NIKHIL

2211CS010561

SECTION-6

Title: Gold Price Prediction using Apache Spark.

Source:https://www.kaggle.com/datasets/sid321axn/gold-price-prediction-dataset

This project demonstrates how to build a machine learning pipeline in PySpark to predict gold prices using historical market data (from FINAL_USO.csv). The model is based on Gradient Boosted Trees (GBT), a powerful ensemble method for regression.

In [8]:
```
from pyspark.sql import SparkSession


spark = SparkSession.builder \
    .appName("GoldPricePrediction") \
    .getOrCreate()
```

Creates a Spark session named "GoldPricePrediction". Reads the CSV file FINAL_USO.csv into a DataFrame. header=True → First row is treated as column names. inferSchema=True → Automatically detects data types (int, double, string). printSchema() → Prints column names and their types. show(5) → Displays the first 5 rows. This is your raw dataset. Likely has columns like Date, Open, High, Low, Close, Volume

In [10]:
```
df = spark.read.option("header", True).option("inferSchema", True).csv("FINAL_US


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

```
root
 |-- Date: date (nullable = true)
 |-- Open: double (nullable = true)
 |-- High: double (nullable = true)
 |-- Low: double (nullable = true)
 |-- Close: double (nullable = true)
 |-- Adj Close: double (nullable = true)
 |-- Volume: integer (nullable = true)
 |-- SP_open: double (nullable = true)
 |-- SP_high: double (nullable = true)
 |-- SP_low: double (nullable = true)
 |-- SP_close: double (nullable = true)
 |-- SP_Ajclose: double (nullable = true)
 |-- SP_volume: integer (nullable = true)
 |-- DJ_open: double (nullable = true)
 |-- DJ_high: double (nullable = true)
 |-- DJ_low: double (nullable = true)
 |-- DJ_close: double (nullable = true)
 |-- DJ_Ajclose: double (nullable = true)
 |-- DJ_volume: integer (nullable = true)
 |-- EG_open: double (nullable = true)
 |-- EG_high: double (nullable = true)
 |-- EG_low: double (nullable = true)
 |-- EG_close: double (nullable = true)
 |-- EG_Ajclose: double (nullable = true)
 |-- EG_volume: integer (nullable = true)
 |-- EU_Price: double (nullable = true)
 |-- EU_open: double (nullable = true)
 |-- EU_high: double (nullable = true)
 |-- EU_low: double (nullable = true)
 |-- EU_Trend: integer (nullable = true)
 |-- OF_Price: double (nullable = true)
 |-- OF_Open: double (nullable = true)
 |-- OF_High: double (nullable = true)
 |-- OF_Low: double (nullable = true)
 |-- OF_Volume: integer (nullable = true)
 |-- OF_Trend: integer (nullable = true)
 |-- OS_Price: double (nullable = true)
 |-- OS_Open: double (nullable = true)
 |-- OS_High: double (nullable = true)
 |-- OS_Low: double (nullable = true)
 |-- OS_Trend: integer (nullable = true)
 |-- SF_Price: integer (nullable = true)
 |-- SF_Open: integer (nullable = true)
 |-- SF_High: integer (nullable = true)
 |-- SF_Low: integer (nullable = true)
 |-- SF_Volume: integer (nullable = true)
 |-- SF_Trend: integer (nullable = true)
 |-- USB_Price: double (nullable = true)
 |-- USB_Open: double (nullable = true)
 |-- USB_High: double (nullable = true)
 |-- USB_Low: double (nullable = true)
 |-- USB_Trend: integer (nullable = true)
 |-- PLT_Price: double (nullable = true)
 |-- PLT_Open: double (nullable = true)
 |-- PLT_High: double (nullable = true)
 |-- PLT_Low: double (nullable = true)
 |-- PLT_Trend: integer (nullable = true)
 |-- PLD_Price: double (nullable = true)
 |-- PLD_Open: double (nullable = true)
```

```
 |-- PLD_High: double (nullable = true)
 |-- PLD_Low: double (nullable = true)
 |-- PLD_Trend: integer (nullable = true)
 |-- RHO_PRICE: integer (nullable = true)
 |-- USDI_Price: double (nullable = true)
 |-- USDI_Open: double (nullable = true)
 |-- USDI_High: double (nullable = true)
 |-- USDI_Low: double (nullable = true)
 |-- USDI_Volume: integer (nullable = true)
 |-- USDI_Trend: integer (nullable = true)
 |-- GDX_Open: double (nullable = true)
 |-- GDX_High: double (nullable = true)
 |-- GDX_Low: double (nullable = true)
 |-- GDX_Close: double (nullable = true)
 |-- GDX_Adj Close: double (nullable = true)
 |-- GDX_Volume: integer (nullable = true)
 |-- USO_Open: double (nullable = true)
 |-- USO_High: double (nullable = true)
 |-- USO_Low: double (nullable = true)
 |-- USO_Close: double (nullable = true)
 |-- USO_Adj Close: double (nullable = true)
 |-- USO_Volume: integer (nullable = true)

+----------+----------+----------+------------------+----------+----------+------
--+----------+----------+----------+----------+------------------+---------+-----
------+-----------+-----------+-----------+-----------+---------+---------+------
---+---------+---------+-----------------+---------+--------+------------------+-
------+------+--------+--------+-------+-------+------+---------+--------+-------
-+-------+-------+------+--------+--------+-------+-------+------+---------+-----
---+---------+--------+--------+-------+---------+---------+--------+--------+---
----+---------+---------+--------+--------+-------+---------+---------+----------
-------+---------+---------+--------+-----------+----------+---------+---------+-
--------+---------+-------------+----------+------------------+------------------
+---------+------------------+------------------+----------+
|      Date|      Open|      High|               Low|     Close| Adj Close|  Volu
me|   SP_open|   SP_high|    SP_low|  SP_close|        SP_Ajclose|SP_volume|    D
J_open|    DJ_high|     DJ_low|   DJ_close| DJ_Ajclose|DJ_volume|  EG_open|  EG_h
igh|   EG_low| EG_close|       EG_Ajclose|EG_volume|EU_Price|           EU_open|E
U_high|EU_low|EU_Trend|OF_Price|OF_Open|OF_High|OF_Low|OF_Volume|OF_Trend|OS_Pric
e|OS_Open|OS_High|OS_Low|OS_Trend|SF_Price|SF_Open|SF_High|SF_Low|SF_Volume|SF_Tr
end|USB_Price|USB_Open|USB_High|USB_Low|USB_Trend|PLT_Price|PLT_Open|PLT_High|PLT
_Low|PLT_Trend|PLD_Price|PLD_Open|PLD_High|PLD_Low|PLD_Trend|RHO_PRICE|       USD
I_Price|USDI_Open|USDI_High|USDI_Low|USDI_Volume|USDI_Trend| GDX_Open| GDX_High|
GDX_Low|GDX_Close|GDX_Adj Close|GDX_Volume|          USO_Open|          USO_High|
USO_Low|         USO_Close|     USO_Adj Close|USO_Volume|
+----------+----------+----------+------------------+----------+----------+------
--+----------+----------+----------+----------+------------------+---------+-----
------+-----------+-----------+-----------+-----------+---------+---------+------
---+---------+---------+-----------------+---------+--------+------------------+-
------+------+--------+--------+-------+-------+------+---------+--------+-------
-+-------+-------+------+--------+--------+-------+-------+------+---------+-----
---+---------+--------+--------+-------+---------+---------+--------+--------+---
----+---------+---------+--------+--------+-------+---------+---------+----------
-------+---------+---------+--------+-----------+----------+---------+---------+-
--------+---------+-------------+----------+------------------+------------------
+---------+------------------+------------------+----------+
|2011-12-15|154.740005|154.949997|151.71000700000005|152.330002|152.330002|215219
00|123.029999|123.199997|121.989998|    122.18|105.44123799999998|199109200|1182
5.29004|11967.83984|11825.21973|11868.80957|11868.80957|136930000|74.550003|76.15
0002|72.150002|72.900002|70.43175500000002|   787900|  1.3018|            1.2982|
```

```
1.3051|1.2957|       1|  105.09| 104.88|  106.5|104.88|    14330|       1|   93.4
2|  94.91|   96.0| 93.33|       0|   53604|  54248|  54248| 52316|   119440|
1|    1.911|   1.911|   1.911|  1.911|        1|  1414.65|  1420.3| 1423.35|1376.
85|        0|   618.85|   614.7|   615.0|  614.6|        1|     1425|80.341000000
00002|   80.565|    80.63|   80.13|      22850|         0|53.009998|53.139999|
51.57|    51.68|    48.973877|  20605600|         36.900002|         36.939999|3
6.049999|         36.130001|         36.130001|  12616700|
|2011-12-16|154.309998|155.369995|        153.899994|155.229996|155.229996|181243
00|122.230003|122.949997|121.300003|121.589996|        105.597549|220481400|   11
870.25|11968.17969|11819.30957|11866.38965|11866.38965|389520000|73.599998|75.099
998|73.349998|74.900002|        72.364037|   896600|  1.3035|1.3019999999999998|
1.3087|1.2997|       1|  103.35| 103.51| 104.56|102.46|   140080|       0|   93.7
9|  93.43|   94.8| 92.53|       1|   53458|  53650|  54030| 52890|    65390|
0|    1.851|   1.851|   1.851|  1.851|        0|  1420.25| 1414.75| 1431.75| 140
0.7|        1|   623.65|   622.6|  623.45|  622.3|        1|     1400|
80.249|   80.175|   80.395|  79.935|      13150|         0|     52.5|    53.18|5
2.040001|    52.68|    49.921513|  16285400|             36.18|               36.5
|    35.73|             36.27|             36.27|  12578800|
|2011-12-19|155.479996|155.860001|        154.360001|154.869995|154.869995|125472
00|122.059998|    122.32|120.029999|120.290001|        104.468536|183903000|1186
6.54004|11925.87988|11735.19043|11766.25977|11766.25977|135170000|69.099998|69.80
0003|64.199997|64.699997|        62.509384|  2096700|  1.2995|            1.3043|
1.3044|1.2981|       0|  103.64| 103.63| 104.57|102.37|   147880|       1|   94.0
9|  93.77|  94.43| 92.55|       1|   52961|  53400|  53400| 52544|    67280|
0|     1.81|    1.81|    1.81|   1.81|        0|   1411.1| 1422.65|  1427.6| 140
4.6|        0|    608.8|   626.0|   630.0|  608.6|        0|     1400|
80.207|     80.3|    80.47|  80.125|        970|         0|52.490002|52.549999|5
1.029999|51.169998|    48.490578|  15120200|36.389998999999996|         36.450001
|    35.93|         36.200001|         36.200001|   7418200|
|2011-12-20|156.820007|157.429993|        156.580002|156.979996|156.979996| 91363
00|    122.18|124.139999|120.370003|    123.93|        107.629784|225418100|1176
9.20996|12117.12988|11768.83008|12103.58008|12103.58008|165180000|66.449997|68.09
9998|     66.0|     67.0|        64.731514|   875300|  1.3079|            1.3003|
1.3133|1.2994|       1|  106.73|  104.3| 107.27|103.91|   170240|       1|   95.5
5|  96.39|   99.7| 96.39|       1|   53487|  52795|  53575| 52595|    55130|
1|    1.927|   1.927|   1.927|  1.927|        1|  1434.75| 1408.95| 1436.55|1408.
15|        1|   626.65|  622.45|  622.45| 622.45|        1|     1400|           8
0.273|    80.89|    80.94|  80.035|      22950|         1|52.380001|    53.25|52.
369999|52.990002|    50.215282|  11644900|         37.299999|37.610001000000004|3
7.220001|         37.560001|         37.560001|  10041600|
|2011-12-21|156.979996|157.529999|        156.130005|157.160004|157.160004|119961
00|    123.93|124.360001|    122.75|124.169998|        107.838242|194230900|1210
3.58008| 12119.7002|11999.44043|12107.74023|12107.74023|163250000|67.099998|69.40
0002|66.900002|     68.5|        66.180725|   837600|  1.3045|            1.3079|
1.3197|1.3024|       0|  107.71| 107.15| 108.17|106.16|   145090|       1|   99.0
1|  97.54|  99.26| 96.81|       1|   53148|  53519|  54184| 52937|    75950|
0|     1.97|    1.97|    1.97|   1.97|        1|  1429.05|  1434.4| 1453.75|1417.
65|        0|    635.9|   625.7|   641.5|  623.8|        1|     1400|
80.35|   80.105|   80.445|   79.55|      24140|         1|53.150002|    53.43|52.
419998|52.959999|    50.186852|   8724300|         37.669998|38.240002000000004|
37.52|38.110001000000004|38.110001000000004|  10728000|
+----------+----------+----------+------------------+----------+----------+------
--+----------+----------+----------+----------+------------------+---------+-----
------+----------+----------+----------+----------+---------+---------+------
---+---------+---------+-----------------+---------+--------+-----------------+-
------+------+--------+--------+-------+-------+------+---------+--------+-------
-+-------+-------+------+--------+--------+-------+-------+------+---------+-----
---+---------+--------+--------+-------+---------+---------+--------+--------+---
----+---------+---------+--------+--------+-------+---------+---------+----------
-------+---------+---------+--------+----------+----------+---------+---------+-
```

```
--------+---------+-------------+----------+-----------------+-----------------
+---------+------------------+-----------------+----------+
only showing top 5 rows
```

Converts the "Date" column from string to actual DateType.

Creates a new column "date_ts" → numeric timestamp (seconds since 1970).

Orders rows chronologically by "Date".

Displays the top 5 rows.

This prepares your data for time series modeling (numerical time input is required).

In [12]:
```
from pyspark.sql.functions import to_date, unix_timestamp, col


df = df.withColumn("Date", to_date(col("Date"), "yyyy-MM-dd"))
df = df.withColumn("date_ts", unix_timestamp(col("Date")).cast("double"))


df = df.orderBy("Date")
df.show(5)
```

```
+----------+----------+----------+------------------+----------+----------+------
--+----------+----------+----------+----------+------------------+---------+-----
------+-----------+-----------+-----------+-----------+---------+---------+------
---+---------+---------+-----------------+---------+--------+------------------+-
------+------+--------+--------+-------+-------+------+---------+--------+-------
-+-------+-------+------+--------+--------+-------+-------+------+---------+-----
---+---------+--------+--------+-------+---------+---------+--------+--------+---
----+---------+---------+--------+--------+-------+---------+---------+----------
-------+---------+---------+--------+-----------+----------+---------+---------+-
--------+---------+-------------+----------+------------------+------------------
+---------+------------------+------------------+----------+-----------+
|      Date|      Open|      High|               Low|     Close| Adj Close|  Volu
me|   SP_open|   SP_high|    SP_low|  SP_close|        SP_Ajclose|SP_volume|    D
J_open|    DJ_high|     DJ_low|   DJ_close| DJ_Ajclose|DJ_volume|  EG_open|  EG_h
igh|   EG_low| EG_close|       EG_Ajclose|EG_volume|EU_Price|           EU_open|E
U_high|EU_low|EU_Trend|OF_Price|OF_Open|OF_High|OF_Low|OF_Volume|OF_Trend|OS_Pric
e|OS_Open|OS_High|OS_Low|OS_Trend|SF_Price|SF_Open|SF_High|SF_Low|SF_Volume|SF_Tr
end|USB_Price|USB_Open|USB_High|USB_Low|USB_Trend|PLT_Price|PLT_Open|PLT_High|PLT
_Low|PLT_Trend|PLD_Price|PLD_Open|PLD_High|PLD_Low|PLD_Trend|RHO_PRICE|       USD
I_Price|USDI_Open|USDI_High|USDI_Low|USDI_Volume|USDI_Trend| GDX_Open| GDX_High|
GDX_Low|GDX_Close|GDX_Adj Close|GDX_Volume|          USO_Open|          USO_High|
USO_Low|         USO_Close|     USO_Adj Close|USO_Volume|    date_ts|
+----------+----------+----------+------------------+----------+----------+------
--+----------+----------+----------+----------+------------------+---------+-----
------+-----------+-----------+-----------+-----------+---------+---------+------
---+---------+---------+-----------------+---------+--------+------------------+-
------+------+--------+--------+-------+-------+------+---------+--------+-------
-+-------+-------+------+--------+--------+-------+-------+------+---------+-----
---+---------+--------+--------+-------+---------+---------+--------+--------+---
----+---------+---------+--------+--------+-------+---------+---------+----------
-------+---------+---------+--------+-----------+----------+---------+---------+-
--------+---------+-------------+----------+------------------+------------------
+---------+------------------+------------------+----------+-----------+
|2011-12-15|154.740005|154.949997|151.71000700000005|152.330002|152.330002|215219
00|123.029999|123.199997|121.989998|    122.18|105.44123799999998|199109200|1182
5.29004|11967.83984|11825.21973|11868.80957|11868.80957|136930000|74.550003|76.15
0002|72.150002|72.900002|70.43175500000002|   787900|  1.3018|            1.2982|
1.3051|1.2957|       1|  105.09| 104.88|  106.5|104.88|    14330|       1|   93.4
2|  94.91|   96.0| 93.33|       0|   53604|  54248|  54248| 52316|   119440|
1|    1.911|   1.911|   1.911|  1.911|        1|  1414.65|  1420.3| 1423.35|1376.
85|        0|   618.85|   614.7|   615.0|  614.6|        1|     1425|80.341000000
00002|   80.565|    80.63|   80.13|      22850|         0|53.009998|53.139999|
51.57|    51.68|    48.973877|  20605600|         36.900002|         36.939999|3
6.049999|         36.130001|         36.130001|  12616700|1.3238874E9|
|2011-12-16|154.309998|155.369995|        153.899994|155.229996|155.229996|181243
00|122.230003|122.949997|121.300003|121.589996|        105.597549|220481400|   11
870.25|11968.17969|11819.30957|11866.38965|11866.38965|389520000|73.599998|75.099
998|73.349998|74.900002|        72.364037|   896600|  1.3035|1.3019999999999998|
1.3087|1.2997|       1|  103.35| 103.51| 104.56|102.46|   140080|       0|   93.7
9|  93.43|   94.8| 92.53|       1|   53458|  53650|  54030| 52890|    65390|
0|    1.851|   1.851|   1.851|  1.851|        0|  1420.25| 1414.75| 1431.75| 140
0.7|        1|   623.65|   622.6|  623.45|  622.3|        1|     1400|
80.249|   80.175|   80.395|  79.935|      13150|         0|     52.5|    53.18|5
2.040001|    52.68|    49.921513|  16285400|             36.18|              36.5
|    35.73|             36.27|             36.27|  12578800|1.3239738E9|
|2011-12-19|155.479996|155.860001|        154.360001|154.869995|154.869995|125472
00|122.059998|    122.32|120.029999|120.290001|        104.468536|183903000|1186
6.54004|11925.87988|11735.19043|11766.25977|11766.25977|135170000|69.099998|69.80
0003|64.199997|64.699997|        62.509384|  2096700|  1.2995|            1.3043|
1.3044|1.2981|       0|  103.64| 103.63| 104.57|102.37|   147880|       1|   94.0
```

```
9|  93.77|  94.43| 92.55|       1|   52961|  53400|  53400| 52544|    67280|
0|     1.81|    1.81|    1.81|   1.81|        0|   1411.1| 1422.65|  1427.6| 140
4.6|        0|    608.8|   626.0|   630.0|  608.6|        0|     1400|
80.207|     80.3|    80.47|  80.125|        970|         0|52.490002|52.549999|5
1.029999|51.169998|    48.490578|  15120200|36.389998999999996|         36.450001
|    35.93|         36.200001|         36.200001|   7418200| 1.324233E9|
|2011-12-20|156.820007|157.429993|        156.580002|156.979996|156.979996| 91363
00|    122.18|124.139999|120.370003|    123.93|        107.629784|225418100|1176
9.20996|12117.12988|11768.83008|12103.58008|12103.58008|165180000|66.449997|68.09
9998|     66.0|     67.0|        64.731514|   875300|  1.3079|            1.3003|
1.3133|1.2994|       1|  106.73|  104.3| 107.27|103.91|   170240|       1|   95.5
5|  96.39|   99.7| 96.39|       1|   53487|  52795|  53575| 52595|    55130|
1|    1.927|   1.927|   1.927|  1.927|        1|  1434.75| 1408.95| 1436.55|1408.
15|        1|   626.65|  622.45|  622.45| 622.45|        1|     1400|           8
0.273|    80.89|    80.94|  80.035|      22950|         1|52.380001|    53.25|52.
369999|52.990002|    50.215282|  11644900|         37.299999|37.610001000000004|3
7.220001|         37.560001|         37.560001|  10041600|1.3243194E9|
|2011-12-21|156.979996|157.529999|        156.130005|157.160004|157.160004|119961
00|    123.93|124.360001|    122.75|124.169998|        107.838242|194230900|1210
3.58008| 12119.7002|11999.44043|12107.74023|12107.74023|163250000|67.099998|69.40
0002|66.900002|     68.5|        66.180725|   837600|  1.3045|            1.3079|
1.3197|1.3024|       0|  107.71| 107.15| 108.17|106.16|   145090|       1|   99.0
1|  97.54|  99.26| 96.81|       1|   53148|  53519|  54184| 52937|    75950|
0|     1.97|    1.97|    1.97|   1.97|        1|  1429.05|  1434.4| 1453.75|1417.
65|        0|    635.9|   625.7|   641.5|  623.8|        1|     1400|
80.35|   80.105|   80.445|   79.55|      24140|         1|53.150002|    53.43|52.
419998|52.959999|    50.186852|   8724300|         37.669998|38.240002000000004|
37.52|38.110001000000004|38.110001000000004|  10728000|1.3244058E9|
+----------+----------+----------+-----------------+----------+----------+------
--+----------+----------+----------+----------+-----------------+---------+-----
------+----------+----------+----------+----------+---------+---------+------
---+---------+---------+-----------------+---------+--------+-----------------+-
------+------+--------+--------+-------+-------+------+---------+--------+-------
-+-------+-------+------+--------+--------+-------+-------+------+---------+-----
---+---------+--------+--------+-------+---------+---------+--------+--------+---
----+---------+---------+--------+--------+-------+---------+---------+----------
-------+---------+---------+--------+----------+----------+---------+---------+-
--------+---------+-------------+----------+-----------------+-----------------
+---------+-----------------+-----------------+----------+-----------+
only showing top 5 rows
```

Defines a window ordered by date. Adds a new column "prev_close" = yesterday's closing price (lag("Close", 1)). Drops rows with NULL values (first row won't have a prev_close). Shows 3 useful columns: Date, Close, and Previous Close. This adds historical dependency (important for time series).

In [14]:

```
from pyspark.sql.window import Window
from pyspark.sql.functions import lag


w = Window.orderBy("Date")
df = df.withColumn("prev_close", lag("Close", 1).over(w))


df = df.na.drop()

df.select("Date", "Close", "prev_close").show(5)
```

```
+----------+------------------+----------+
|      Date|             Close|prev_close|
+----------+------------------+----------+
|2011-12-16|        155.229996|152.330002|
|2011-12-19|        154.869995|155.229996|
|2011-12-20|        156.979996|154.869995|
|2011-12-21|        157.160004|156.979996|
|2011-12-22|156.03999299999995|157.160004|
+----------+------------------+----------+
only showing top 5 rows
```

Defines a window ordered by date. Adds a new column "prev_close" = yesterday's closing price (lag("Close", 1)). Drops rows with NULL values (first row won't have a prev_close). Shows 3 useful columns: Date, Close, and Previous Close. This adds historical dependency (important for time series).

In [17]:
```
from pyspark.ml.feature import VectorAssembler, StandardScaler

# Select features
feature_cols = ["date_ts", "Open", "High", "Low", "Volume", "prev_close"]

# Assemble feature vector
assembler = VectorAssembler(inputCols=feature_cols, outputCol="features_unnormal

# Scale features
scaler = StandardScaler(inputCol="features_unnormalized", outputCol="features",
```

feature_cols: input features for the model. VectorAssembler → combines multiple numeric columns into one vector column (features_unnormalized). StandardScaler → standardizes features (zero mean, unit variance). Helps ML algorithms converge better. Final column is features.

In [19]:
```
from pyspark.ml.regression import GBTRegressor

# Gradient Boosted Trees Regressor
gbt = GBTRegressor(featuresCol="features", labelCol="Close", predictionCol="pred
```

Uses Gradient Boosted Trees for regression. featuresCol="features" → input vector. labelCol="Close" → actual value we want to predict. predictionCol="prediction" → model output.

In [26]:
```
from pyspark.ml import Pipeline

# Create pipeline
pipeline = Pipeline(stages=[assembler, scaler, gbt])
```

A Pipeline chains multiple steps into one process. Stages: Assemble features Scale features Apply Gradient Boosted Trees This makes training and prediction seamless.

In [29]:
```
# Time-aware split is better for time series, but here we use randomSplit
train_df, test_df = df.randomSplit([0.8, 0.2], seed=42)
```

Splits dataset into 80% training, 20% testing.

In [32]:
```
# Fit the model on training data
model = pipeline.fit(train_df)
```

fit() trains the pipeline model on training data.

In [34]:
```
predictions = model.transform(test_df)


predictions.select("Date", "Close", "prediction").show(10)
```

```
+----------+----------+------------------+
|      Date|     Close|        prediction|
+----------+----------+------------------+
|2011-12-20|156.979996|156.29122735384558|
|2011-12-27|154.910004|156.27259409617338|
|2011-12-29|150.339996|150.55665351500136|
|2012-01-06|157.199997| 159.0582981367929|
|2012-01-17|     160.5|161.32714890954537|
|2012-01-23|163.160004| 161.7929685077807|
|2012-02-01|169.559998|168.62801236044072|
|2012-02-09|168.020004|171.32970090561616|
|2012-02-24|172.229996|170.44172246773877|
|2012-02-27|171.699997|168.22654449324068|
+----------+----------+------------------+
only showing top 10 rows
```

transform() applies model to test data, generating predictions. Shows actual vs predicted gold prices for 10 rows.

In [36]:
```
from pyspark.ml.evaluation import RegressionEvaluator

evaluator = RegressionEvaluator(labelCol="Close", predictionCol="prediction", me
rmse = evaluator.evaluate(predictions)

print(f"Root Mean Squared Error (RMSE) on test data = {rmse}")
```

```
Root Mean Squared Error (RMSE) on test data = 0.8265727329134374
```

Uses RegressionEvaluator to measure accuracy. RMSE (Root Mean Squared Error) → standard metric for regression. Lower RMSE = better predictions. Prints model error on test data.