INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

DAY 13

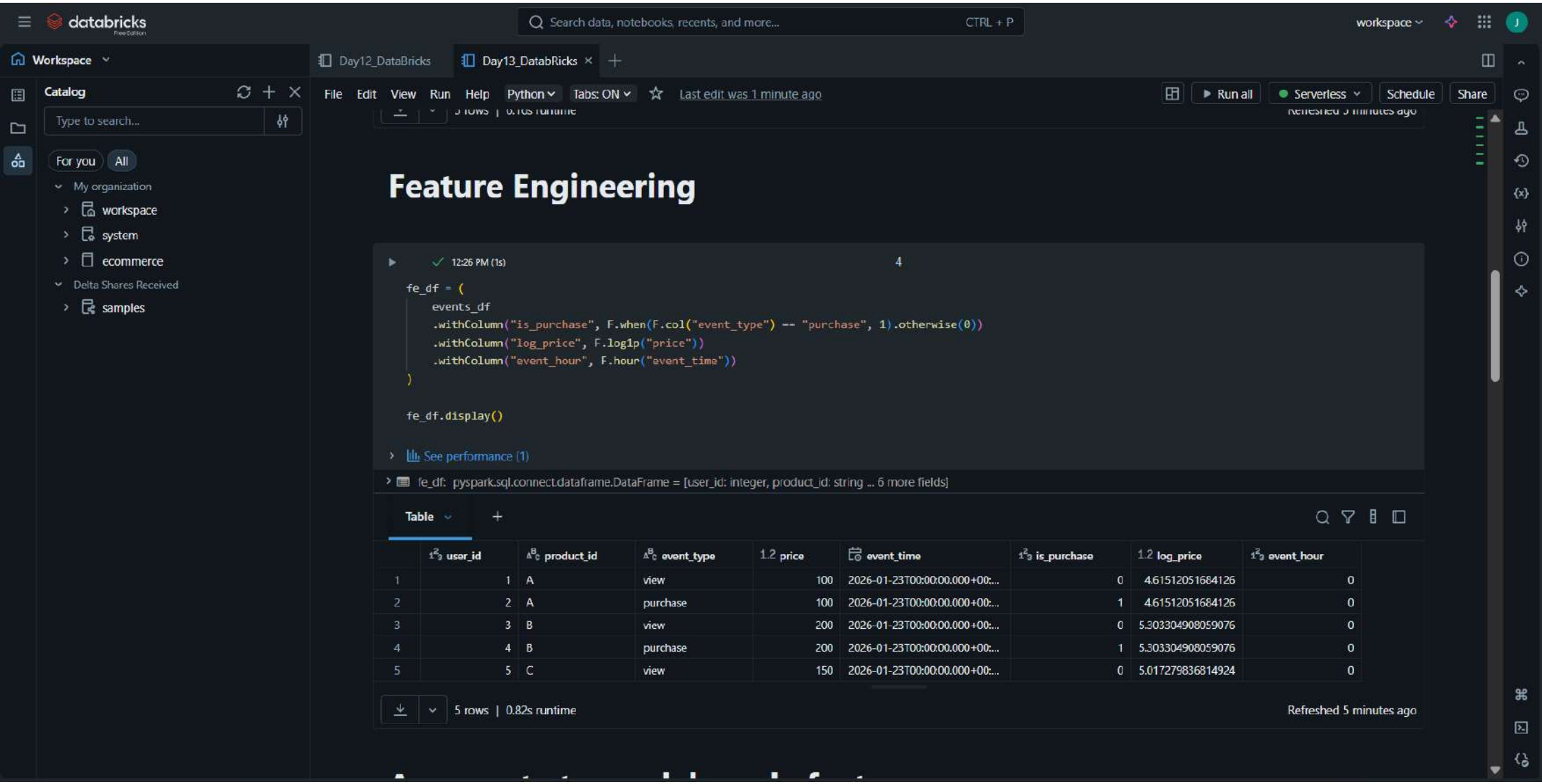
Topic:

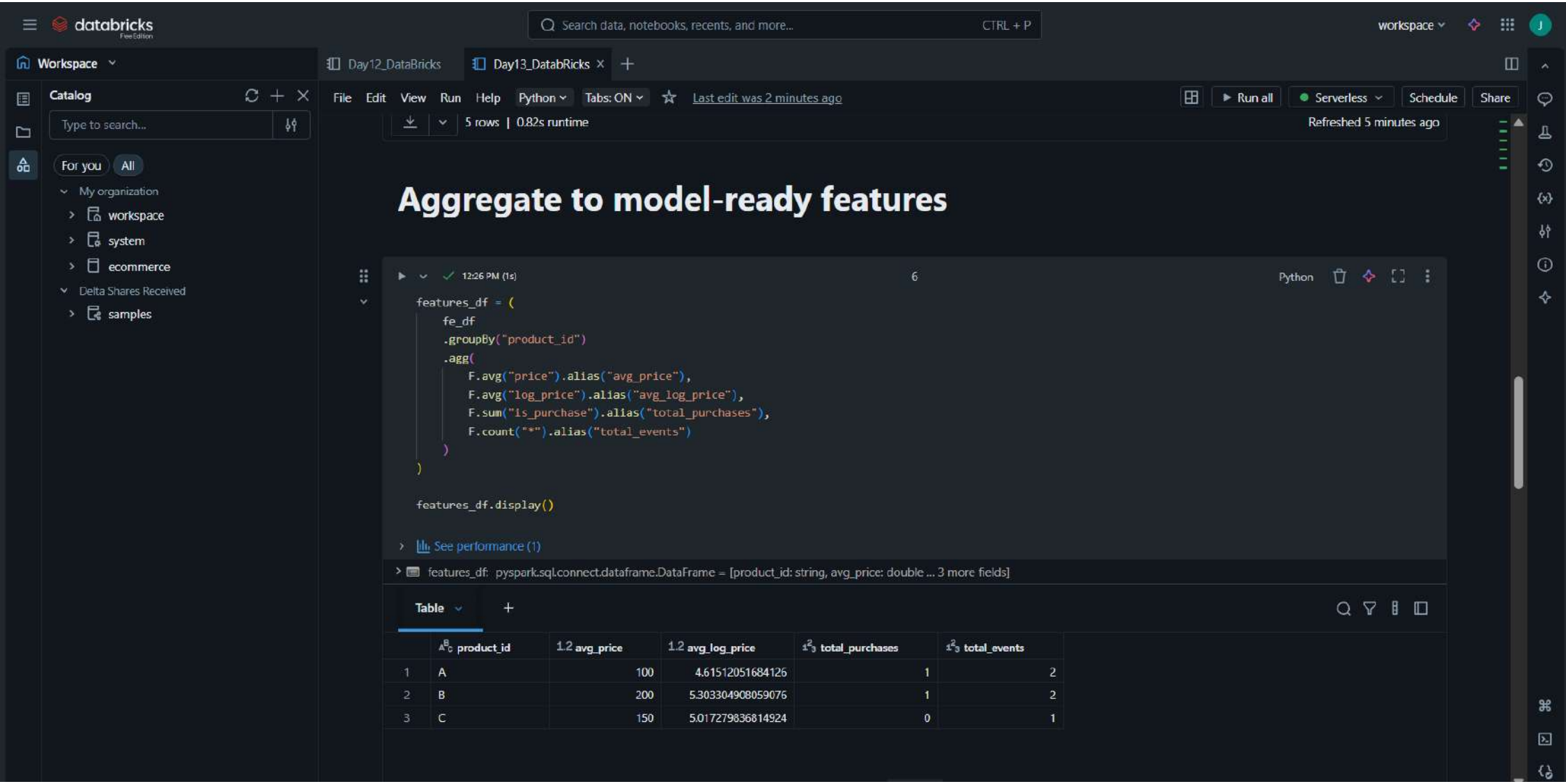
Model Comparison & Feature Engineering

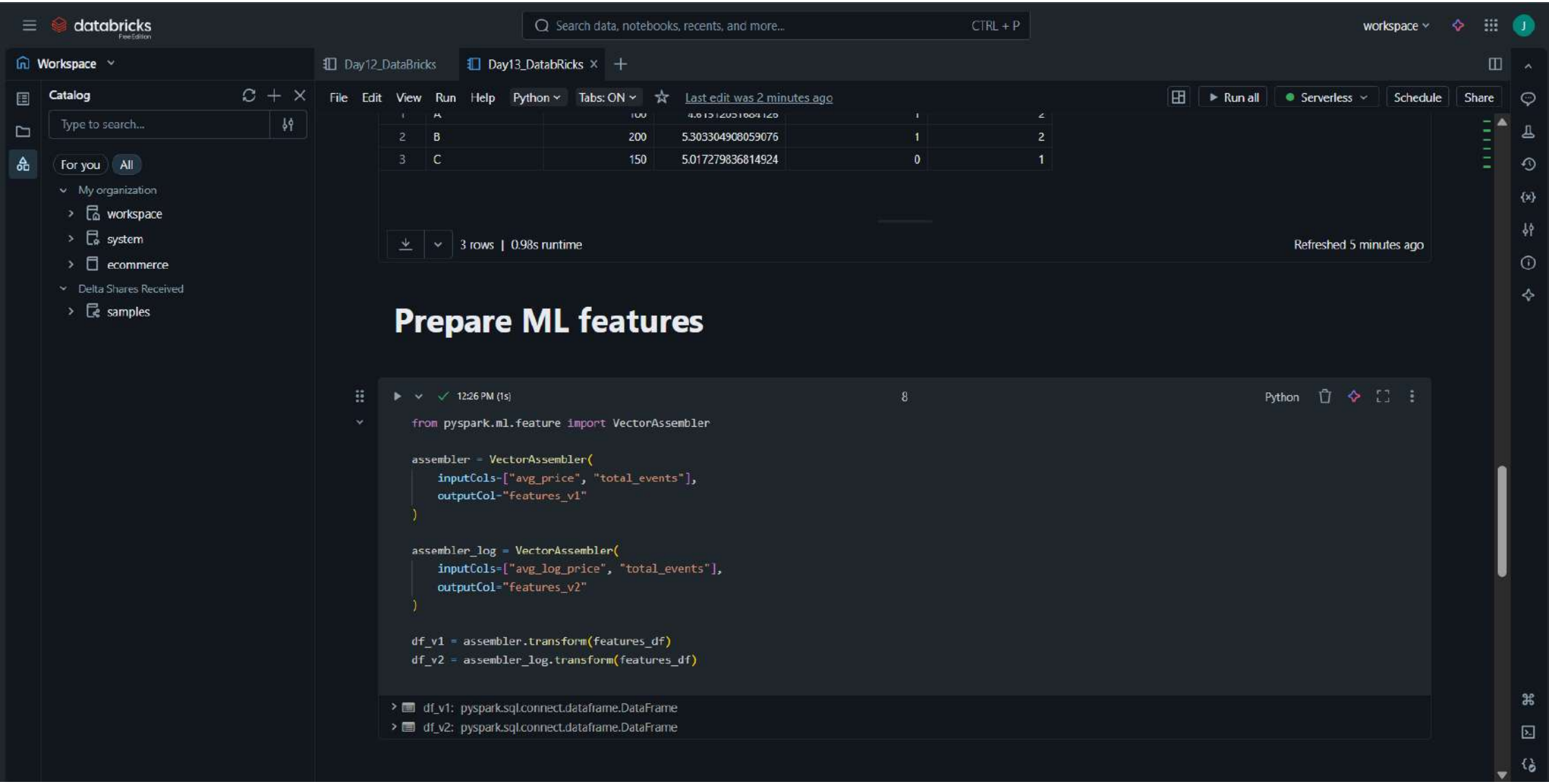
Challenge:

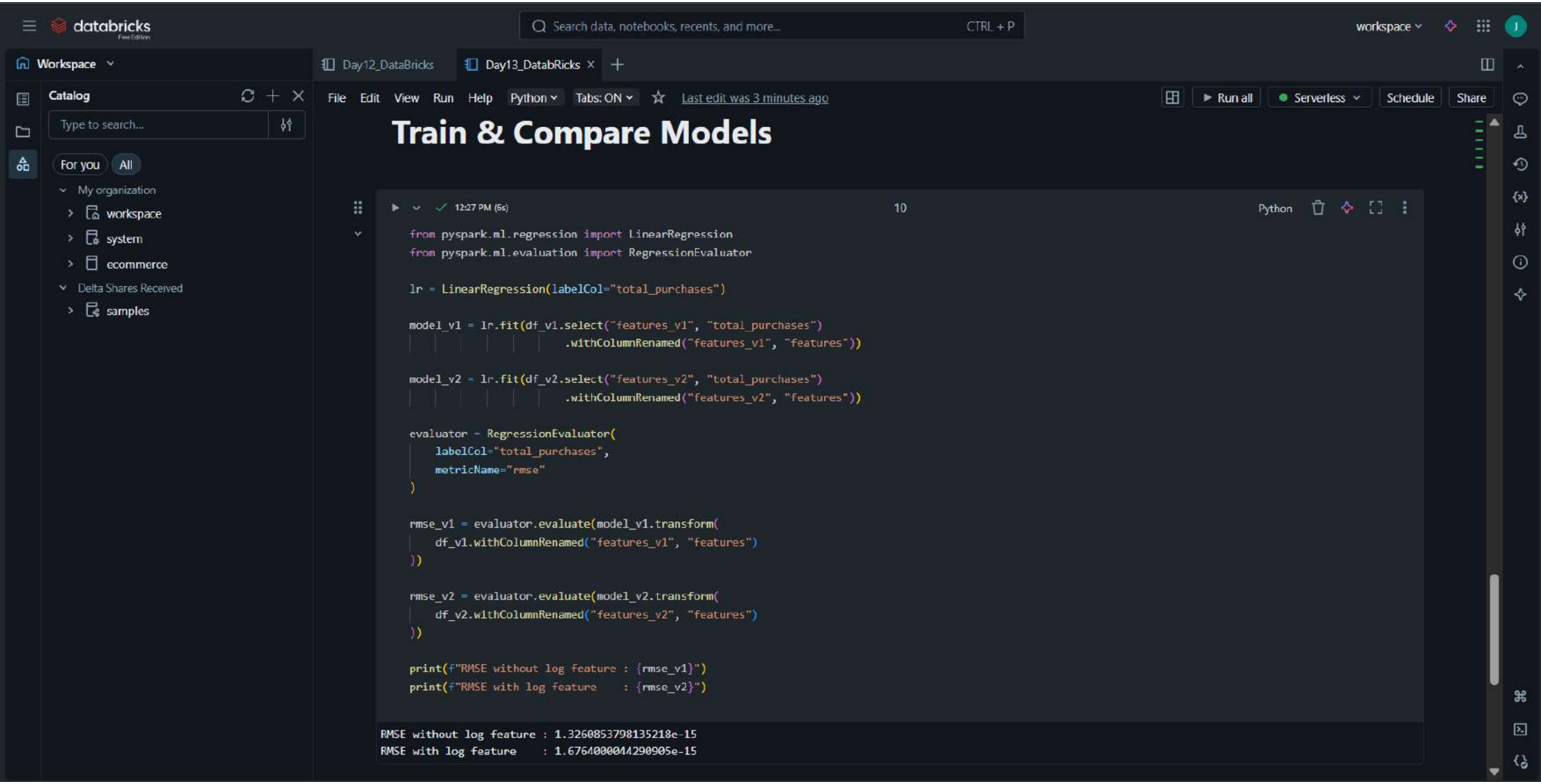
1. Train 3 different models
2. Compare metrics in MLflow
3. Build Spark ML pipeline
4. Select best model

#DatabricksWithIDC


databricks
Free Edition
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day12_DataBricks
Day13_DatabRicks
Catalog
Type to search...
For you
All
My organization
workspace
system
ecommerce
Delta Shares Received
samples
File Edit View Run Help Python Tabs: ON
Last edit was 1 minute ago
Run all
Serverless
Schedule
Share
Feature Engineering
12:26 PM (1s)
4
fe_df = (
events_df
.withColumn("is_purchase", F.when(F.col("event_type") == "purchase", 1).otherwise(0))
.withColumn("log_price", F.log1p("price"))
.withColumn("event_hour", F.hour("event_time"))
)
fe_df.display()
See performance (1)
fe_df: pyspark.sql.connect.dataframe.DataFrame = [user_id: integer, product_id: string ... 6 more fields]
Table
user_id
product_id
event_type
price
event_time
is_purchase
log_price
event_hour
1 1 A view 100 2026-01-23T00:00:00.000+00:... 0 4.61512051684126 0
2 2 A purchase 100 2026-01-23T00:00:00.000+00:... 1 4.61512051684126 0
3 3 B view 200 2026-01-23T00:00:00.000+00:... 0 5.303304908059076 0
4 4 B purchase 200 2026-01-23T00:00:00.000+00:... 1 5.303304908059076 0
5 5 C view 150 2026-01-23T00:00:00.000+00:... 0 5.017279836814924 0
5 rows | 0.82s runtime
Refreshed 5 minutes ago


databricks
Search data, notebooks, recents, and more... CTRL + P
workspace
Workspace
Day12_DataBricks
Day13_DatabRicks
Catalog
Type to search...
For you
All
My organization
workspace
system
ecommerce
Delta Shares Received
samples
File Edit View Run Help Python Tabs: ON Last edit was 2 minutes ago
Run all
Serverless
Schedule
Share
5 rows | 0.82s runtime
Refreshed 5 minutes ago
Aggregate to model-ready features
12:26 PM (1s)
6
Python
features_df = (
fe_df
.groupBy("product_id")
.agg(
F.avg("price").alias("avg_price"),
F.avg("log_price").alias("avg_log_price"),
F.sum("is_purchase").alias("total_purchases"),
F.count("*").alias("total_events")
)
)
features_df.display()
See performance (1)
features_df: pyspark.sql.connect.dataframe.DataFrame = [product_id: string, avg_price: double ... 3 more fields]
Table
product_id
avg_price
avg_log_price
total_purchases
total_events
1 A 100 4.61512051684126 1 2
2 B 200 5.303304908059076 1 2
3 C 150 5.017279836814924 0 1


databricks
Free Edition
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day12_DataBricks
Day13_DatabRicks
Catalog
Type to search...
For you
All
My organization
workspace
system
ecommerce
Delta Shares Received
samples
File Edit View Run Help Python Tabs: ON
Last edit was 2 minutes ago
Run all
Serverless
Schedule
Share
2 B 200 5.303304908059076 1 2
3 C 150 5.017279836814924 0 1
3 rows | 0.98s runtime
Refreshed 5 minutes ago
Prepare ML features
12:26 PM (1s)
Python
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(
inputCols=["avg_price", "total_events"],
outputCol="features_v1"
)
assembler_log = VectorAssembler(
inputCols=["avg_log_price", "total_events"],
outputCol="features_v2"
)
df_v1 = assembler.transform(features_df)
df_v2 = assembler_log.transform(features_df)
df_v1: pyspark.sql.connect.dataframe.DataFrame
df_v2: pyspark.sql.connect.dataframe.DataFrame


databricks
Free Edition
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day12_DataBricks
Day13_DatabRicks
Catalog
Type to search...
For you
All
My organization
workspace
system
ecommerce
Delta Shares Received
samples
File Edit View Run Help Python Tabs: ON
Last edit was 3 minutes ago
Run all
Serverless
Schedule
Share
Train & Compare Models
12:27 PM (6s)
10
Python
from pyspark.ml.regression import LinearRegression
from pyspark.ml.evaluation import RegressionEvaluator
lr = LinearRegression(labelCol="total_purchases")
model_v1 = lr.fit(df_v1.select("features_v1", "total_purchases")
.withColumnRenamed("features_v1", "features"))
model_v2 = lr.fit(df_v2.select("features_v2", "total_purchases")
.withColumnRenamed("features_v2", "features"))
evaluator = RegressionEvaluator(
labelCol="total_purchases",
metricName="rmse"
)
rmse_v1 = evaluator.evaluate(model_v1.transform(
df_v1.withColumnRenamed("features_v1", "features")
))
rmse_v2 = evaluator.evaluate(model_v2.transform(
df_v2.withColumnRenamed("features_v2", "features")
))
print(f"RMSE without log feature : {rmse_v1}")
print(f"RMSE with log feature : {rmse_v2}")
RMSE without log feature : 1.3260853798135218e-15
RMSE with log feature : 1.6764000044290905e-15