INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

DAY 13

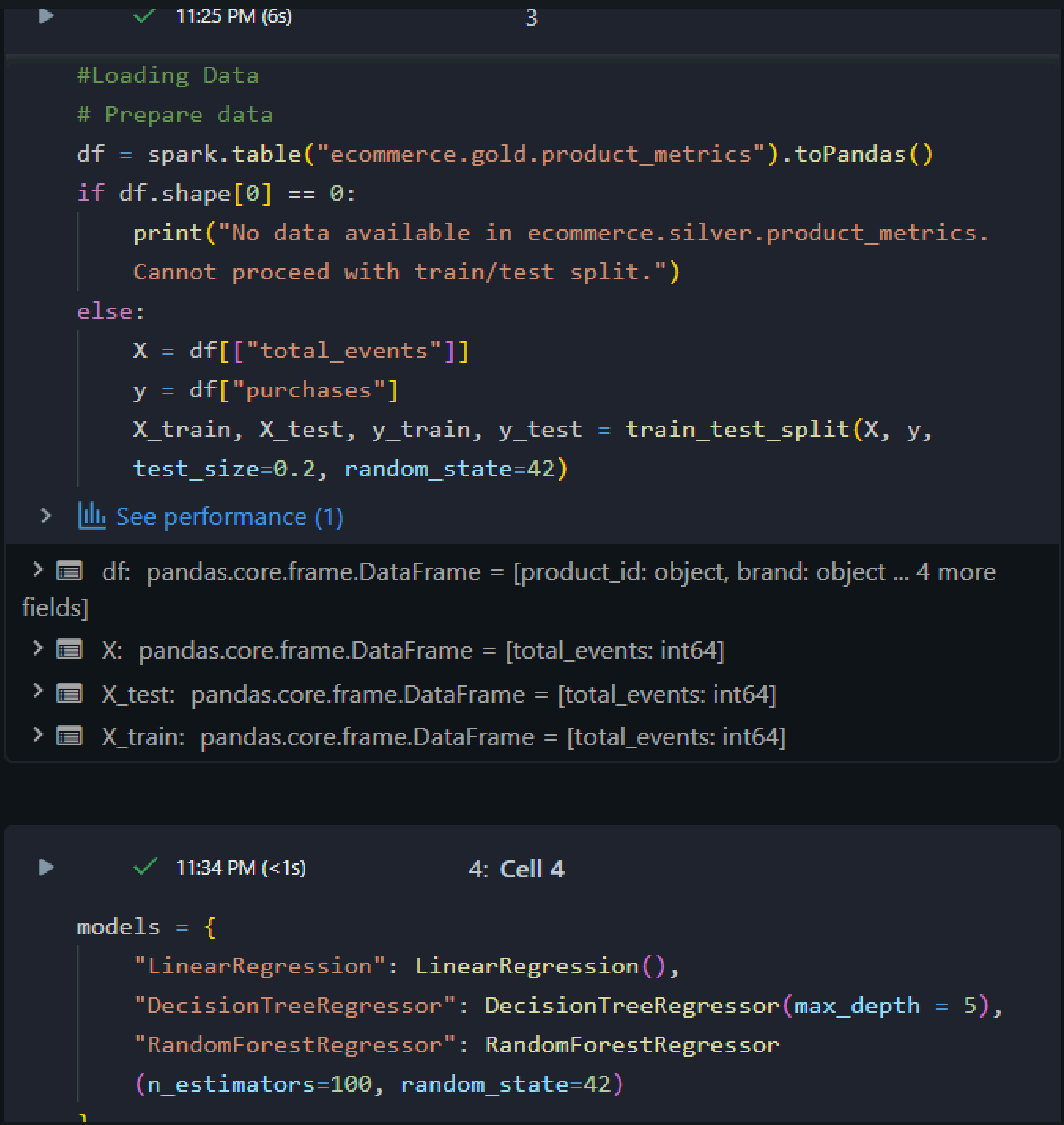
Topic:

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



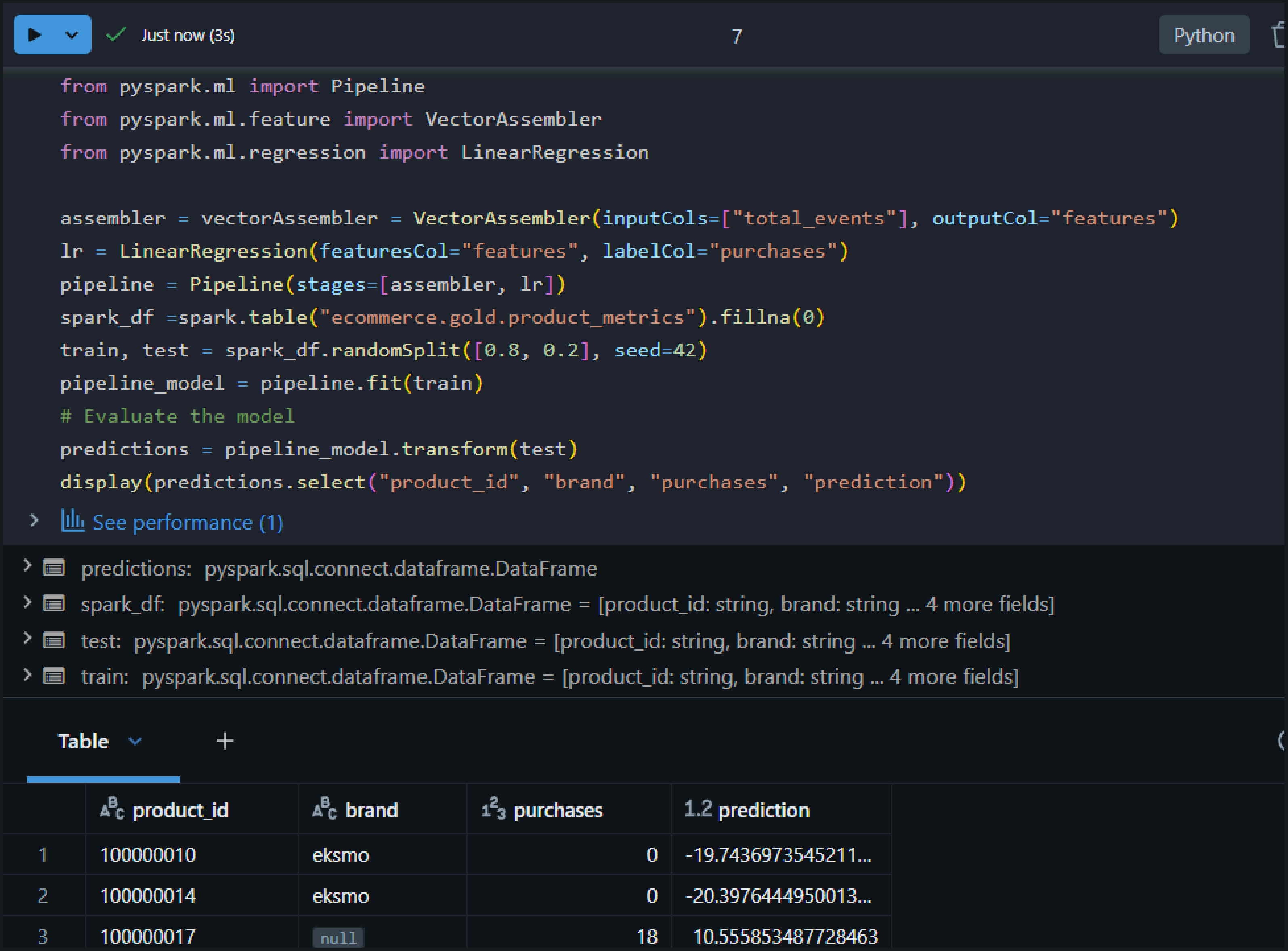
▶ ✓ 11:25 PM (6s) 3

```
#Loading Data
# Prepare data
df = spark.table("ecommerce.gold.product_metrics").toPandas()
if df.shape[0] == 0:
    print("No data available in ecommerce.silver.product_metrics. Cannot proceed with train/test split.")
else:
    X = df[["total_events"]]
    y = df["purchases"]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
› df:  pandas.core.frame.DataFrame = [product_id: object, brand: object ... 4 more fields]
› X:  pandas.core.frame.DataFrame = [total_events: int64]
› X_test:  pandas.core.frame.DataFrame = [total_events: int64]
› X_train:  pandas.core.frame.DataFrame = [total_events: int64]
```

▶ ✓ 11:34 PM (<1s) 4: Cell 4

```
models = {
    "LinearRegression": LinearRegression(),
    "DecisionTreeRegressor": DecisionTreeRegressor(max_depth = 5),
    "RandomForestRegressor": RandomForestRegressor(n_estimators=100, random_state=42)
}
```



```python
from pyspark.ml import Pipeline
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.regression import LinearRegression

assembler = vectorAssembler = VectorAssembler(inputCols=["total_events"], outputCol="features")
lr = LinearRegression(featuresCol="features", labelCol="purchases")
pipeline = Pipeline(stages=[assembler, lr])
spark_df =spark.table("ecommerce.gold.product_metrics").fillna(0)
train, test = spark_df.randomSplit([0.8, 0.2], seed=42)
pipeline_model = pipeline.fit(train)
# Evaluate the model
predictions = pipeline_model.transform(test)
display(predictions.select("product_id", "brand", "purchases", "prediction"))
```

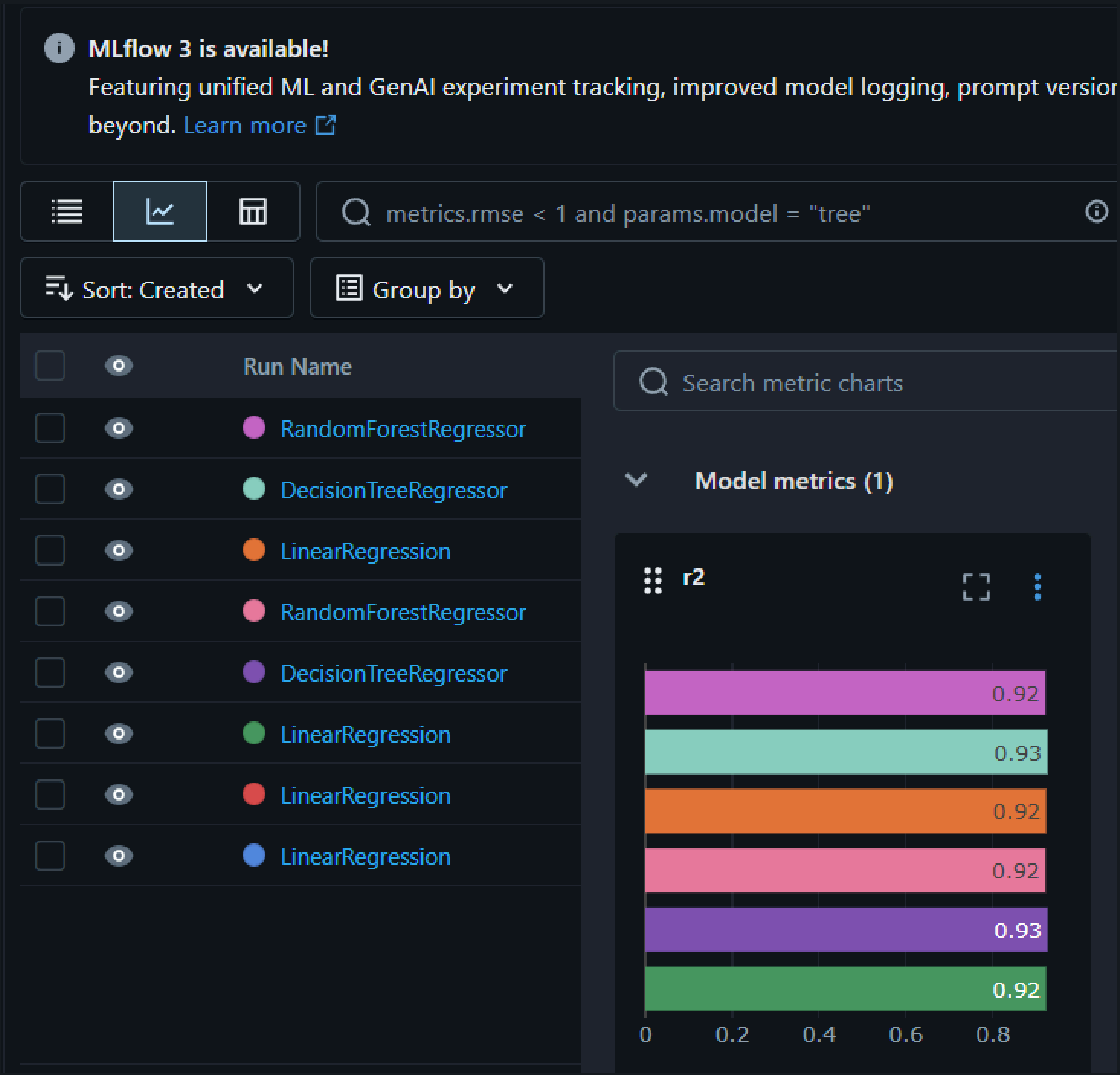
predictions: pyspark.sql.connect.dataframe.DataFrame

spark_df: pyspark.sql.connect.dataframe.DataFrame = [product_id: string, brand: string ... 4 more fields]

test: pyspark.sql.connect.dataframe.DataFrame = [product_id: string, brand: string ... 4 more fields]

train: pyspark.sql.connect.dataframe.DataFrame = [product_id: string, brand: string ... 4 more fields]

	product_id	brand	purchases	prediction
1	100000010	eksmo	0	-19.7436973545211...
2	100000014	eksmo	0	-20.3976444950013...
3	100000017	null	18	10.555853487728463



MLflow 3 is available!
Featuring unified ML and GenAI experiment tracking, improved model logging, prompt versior
beyond. Learn more

metrics.rmse < 1 and params.model = "tree"

Sort: Created

Group by

□		Run Name
□		● RandomForestRegressor
□		● DecisionTreeRegressor
□		● LinearRegression
□		● RandomForestRegressor
□		● DecisionTreeRegressor
□		● LinearRegression
□		● LinearRegression
□		● LinearRegression

Search metric charts

Model metrics (1)