INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

DAY 12

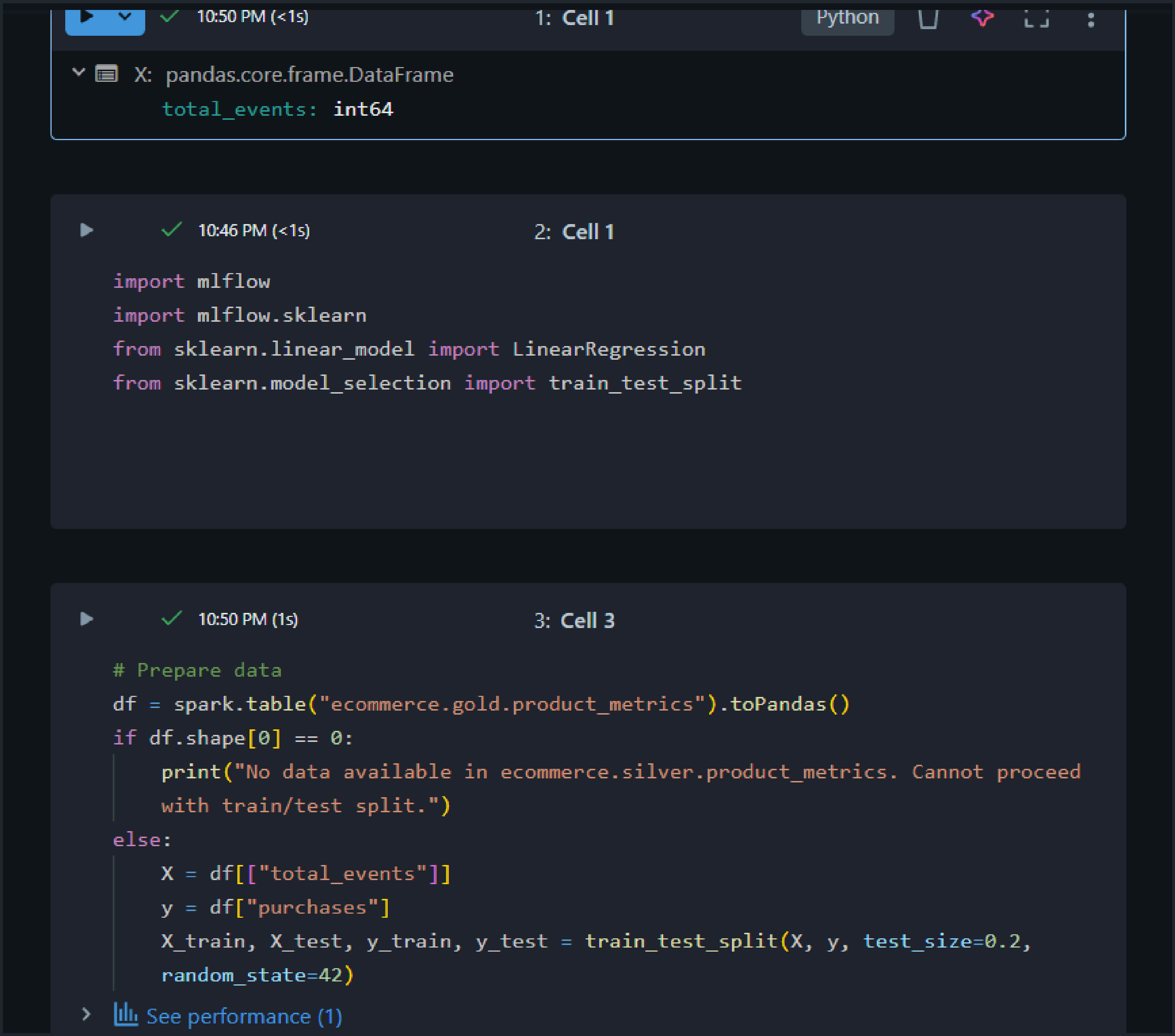
Topic:

MLflow Basics

Challenge:

1. Train simple regression model
2. Log parameters, metrics, model
3. View in MLflow UI
4. Compare runs

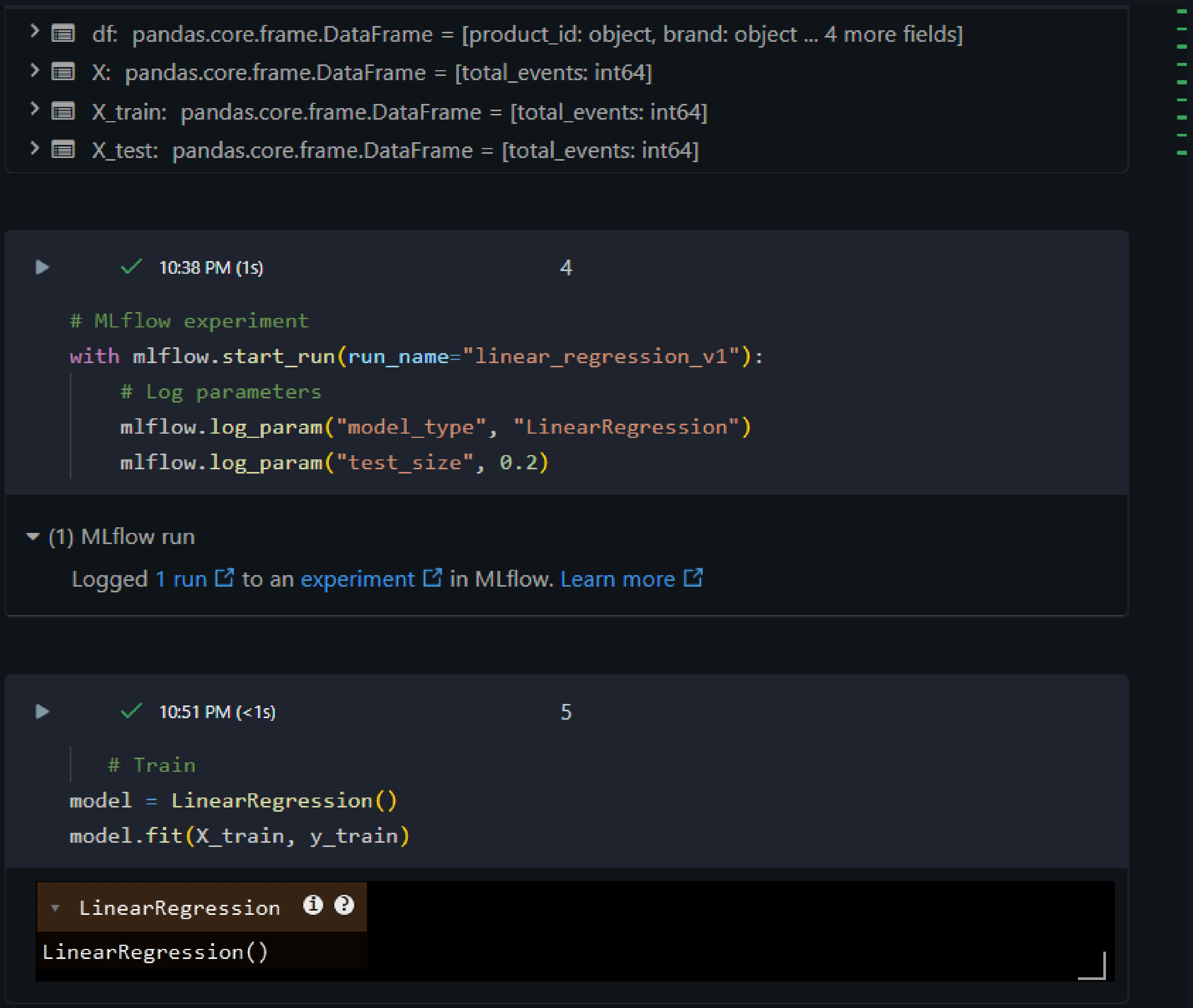
#DatabricksWithIDC



```
X: pandas.core.frame.DataFrame
    total_events: int64
```

```python
import mlflow
import mlflow.sklearn
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

```python
# 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_train:  pandas.core.frame.DataFrame = [total_events: int64]
X_test:  pandas.core.frame.DataFrame = [total_events: int64]
```

✓ 10:38 PM (1s) 4

```
# MLflow experiment
with mlflow.start_run(run_name="linear_regression_v1"):
    # Log parameters
    mlflow.log_param("model_type", "LinearRegression")
    mlflow.log_param("test_size", 0.2)
```

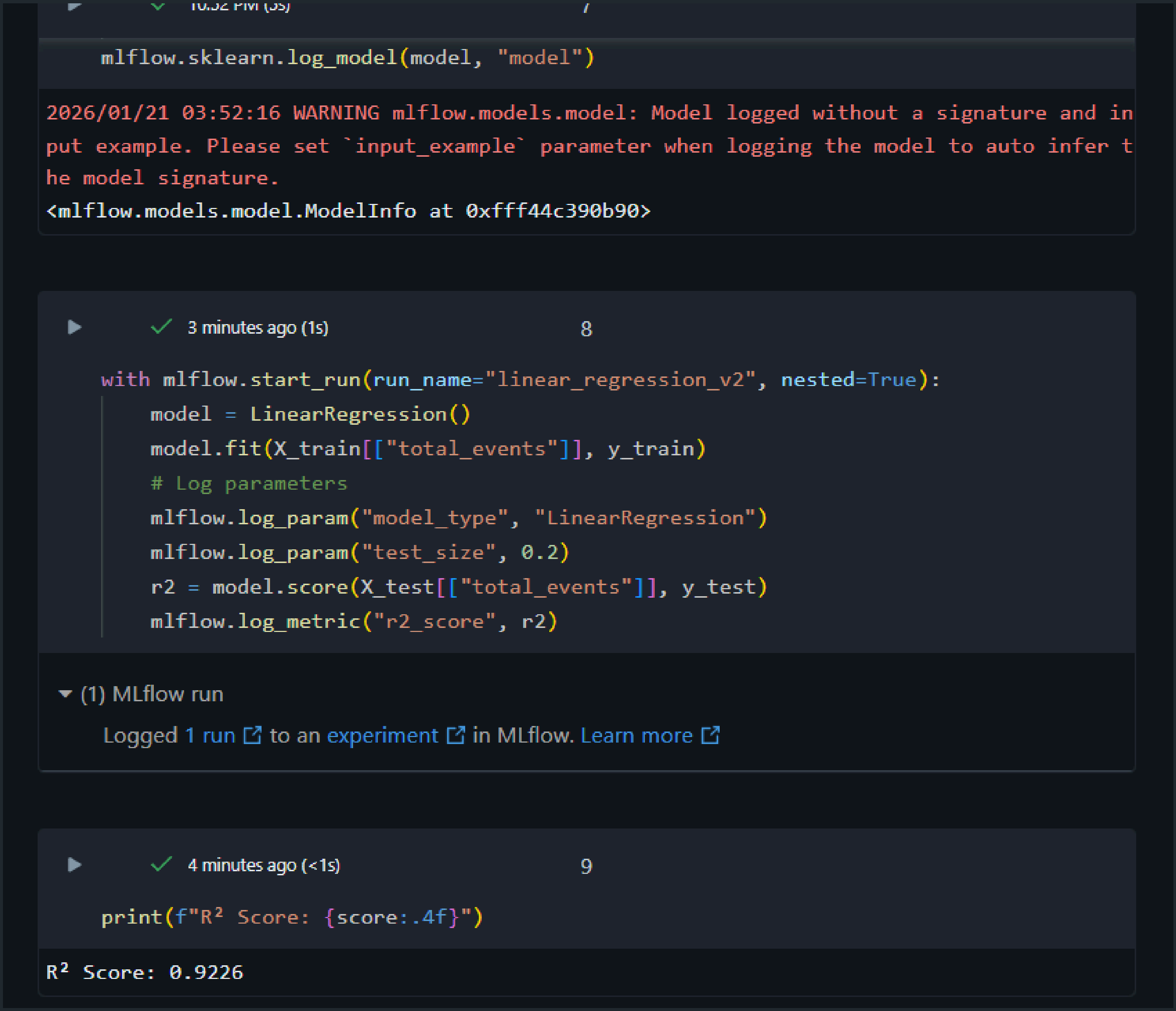
(1) MLflow run

Logged 1 run to an experiment in MLflow. Learn more

✓ 10:51 PM (<1s) 5

```
    # Train
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```



10:52 PM (3s) 7

```python
    mlflow.sklearn.log_model(model, "model")
```

```
2026/01/21 03:52:16 WARNING mlflow.models.model: Model logged without a signature and in
put example. Please set `input_example` parameter when logging the model to auto infer t
he model signature.
<mlflow.models.model.ModelInfo at 0xfff44c390b90>
```

3 minutes ago (1s) 8

```python
with mlflow.start_run(run_name="linear_regression_v2", nested=True):
    model = LinearRegression()
    model.fit(X_train[["total_events"]], y_train)
    # Log parameters
    mlflow.log_param("model_type", "LinearRegression")
    mlflow.log_param("test_size", 0.2)
    r2 = model.score(X_test[["total_events"]], y_test)
    mlflow.log_metric("r2_score", r2)
```

(1) MLflow run

Logged 1 run to an experiment in MLflow. Learn more

4 minutes ago (<1s) 9

```python
print(f"R² Score: {score:.4f}")
```

```
R² Score: 0.9226
```