

Module 11: End-to-End Hybrid Pipeline Integration

Merging Day 1 Unstructured LLM Extractions with Day 2 Supervised Machine

Learning Datasets & Execution

Hybrid Architecture Objectives



1. Dataset Ingestion

Import external training datasets combining raw textual user feedback alongside traditional numerical telemetry data into your Python workflow.



2. LLM Feature Extraction

Transform qualitative text fields into quantitative features (e.g., sentiment probabilities, entity flags) ready for downstream execution.



3. Supervised Execution

Feed unified feature arrays directly into Scikit-Learn estimators to compute automated, high-precision business decisions.

System Architecture Workflow

Dataset Import

Load raw enterprise CSV files containing unstructured feedback and structured metrics using Pandas.

Scikit-Learn Execution

Execute data splits, fit supervised classifiers like RandomForestClassifier, and validate results.

600 × 400

Execution Step 1: Importing Datasets

```
import pandas as pd from sklearn.model_selection import
train_test_split
# 1. Import dataset containing text and structured
telemetry
df =
pd.read_csv('customer_feedback_pipeline.csv')
# 2. Isolate customer feedback pipeline numerical inputs
X =
df[['extracted_sentiment', 'numerical_telemetry']] y =
df['business_target']
# 3. Create train test splits for robust verification
X_train, X_test, y_train, y_test =
train_test_split( X, y, test_size=0.2, random_state=42 )
```

Data Loading Strategy

We import structured logs alongside text-derived variables using standard Pandas dataframe readers, preparing clean matrix partitions for the training phase.

Execution Step 2: Model Training & Predict

```
from sklearn.ensemble import RandomForestClassifier from
sklearn.metrics import accuracy_score
# 1. Initialize import sklearn supervised algorithm
model =
RandomForestClassifier( n_estimators=100, random_state=42 )
# 2. Fit model using the hybrid feature matrix
model.fit(X_train,
y_train) # 3. Execute final model predictions on test data
predictions = model.predict(X_test) score =
accuracy_score(y_test, predictions) print(f
Hybrid Pipeline Accuracy: {score:.2f}
)
```

Supervised Execution

The code invokes RandomForestClassifier.fit() to learn decision rules across our hybrid data features, yielding production-ready predictive metrics.

Module Deliverables & Outcomes

- ✓ **Automated Dataset Ingestion:** Python scripts capable of loading external data inputs reliably into unified analysis tables.
- ✓ **Robust Execution Logic:** Clean Scikit-Learn implementations utilizing `fit()` and evaluation checks.
- ✓ **Production Hybrid Pipeline:** Complete bridge connecting generative LLM outputs with deterministic predictive models.