Evidently AI Monitoring and Data Drift

Evidently AI provides a flexible and powerful way to monitor machine learning models in production. It offers a suite of metrics for model performance, data drift, and target drift, along with visualization tools.

Steps for Classification Model Monitoring with Evidently AI:

1. **Prepare Your Data**:

   Ensure you have both reference and current datasets (typically, training and production data).

2. **Train Your Classification Model**:

   Train your model using any ML library (scikit-learn, TensorFlow, etc.). Example with RandomForest:

   ```python
   from sklearn.ensemble import RandomForestClassifier
   clf = RandomForestClassifier()
   clf.fit(X_train, y_train)
   ```

3. **Generate Predictions**:

   After training, generate predictions on the test or production data.

4. **Define Column Mapping**:

   Map your dataset columns for Evidently to distinguish between features.

   Example:

```python
column_mapping = {
    "target": "actual_label",
    "prediction": "predicted_label",
    "numerical_features": ["age", "salary"],
    "categorical_features": ["gender", "department"]
}
```

5. **Create Evidently Classification Report**:
   Use the `ClassificationPreset` to evaluate model performance metrics like Accuracy, Precision, Recall:

```python
from evidently import Report
from evidently.metrics import ClassificationPreset

report = Report(metrics=[ClassificationPreset()])
report.run(reference_data=train_data, current_data=test_data, column_mapping=column_mapping)
report.show()
```

Steps for Monitoring Data Drift:

1. **Prepare Reference and Current Data**:

   Reference data is typically the training dataset, and current data represents new incoming data.

2. **Define Column Mapping**:

   Specify which columns are numerical and categorical.

   Example:

   ```python

   column_mapping = {

       "numerical_features": ["age", "salary"],

       "categorical_features": ["gender", "department"]

   }

   ```

3. **Set Up Data Drift Detection**:

   Use Evidently's `DataDriftPreset` to detect and visualize drift.

   ```python

   from evidently import Report

   from evidently.metrics import DataDriftPreset

   report = Report(metrics=[DataDriftPreset()])

   report.run(reference_data=reference_data, current_data=current_data, column_mapping=column_mapping)

   report.show()

   ```

4. **Customize Thresholds (Optional)**:

You can adjust the sensitivity by setting a custom threshold:

```python
report = Report(metrics=[DataDriftPreset(threshold=0.05)])
```

5. **Monitor Automatically**:

Set up regular monitoring in your MLOps pipeline to continuously check for data drift.

Additional Metrics Available in Evidently AI:

- **Performance Metrics** (for classification and regression).
- **Data Drift**: Detects shifts in feature distributions.
- **Prediction Drift**: Monitors shifts in the distribution of model predictions.
- **Target Drift**: Identifies changes in the distribution of the target variable.
- **Data Quality**: Ensures data completeness and consistency over time.