Nielsen IQ Label Insight Migration Guide

Data Classification & Attribution Tasks

Overview

Nielsen IQ's Label Insight platform uses AI/ML algorithms to automatically analyze product data and generate structured attributes. This document outlines the migration tasks and classification processes involved.

Understanding Label Insight Classification

What Label Insight Actually Does

```
Raw Product Data → AI Processing → Structured Attributes → Business Intelligence
```

Input Sources:

- Product packaging images
- Ingredient lists (text)
- Marketing claims and descriptions
- Nutritional information
- Certification logos and text
- Product names and brand information

Output Attributes:

- Dietary classifications (Organic, Vegan, Gluten-Free)
- Health claims (High Protein, Low Sodium, Heart Healthy)
- Clean label indicators (No Artificial Colors, Natural)
- Sourcing attributes (Sustainable, Local, Premium)
- Certification mappings (USDA Organic, Non-GMO Project)

Migration Task Breakdown

Phase 1: Data Extraction & Preparation (2-3 weeks)

Task 1.1: Legacy Data Audit

```javascript
// Inventory existing Nielsen data
const dataAudit = {
  productCatalog: {
    totalProducts: 847293,
    withImages: 432156,     // 51% have packaging images
    withIngredients: 678234, // 80% have ingredient lists
    withNutrition: 720145,   // 85% have nutrition facts
    withClaims: 234567       // 28% have marketing claims
  },
  dataQuality: {
    completeRecords: 45,    // Only 45% have all required fields
    imageQuality: 'mixed',  // Varying resolution and clarity
    textAccuracy: 78        // 78% of text data is clean
  }
};
```

Action Items:

- Catalog all existing product data sources
- Identify data quality issues
- Map current attribute schema to Nielsen IQ taxonomy
- Prioritize products by completeness and business importance

Task 1.2: Data Cleaning & Standardization

```python
# Example data cleaning pipeline
def clean_product_data(raw_product):
    cleaned = {
        'upc': standardize_upc(raw_product.upc),
        'name': clean_product_name(raw_product.name),
        'brand': normalize_brand_name(raw_product.brand),
        'category': map_to_nielsen_taxonomy(raw_product.category),
        'ingredients': parse_ingredient_list(raw_product.ingredients),
        'images': validate_and_resize_images(raw_product.images),
        'claims': extract_marketing_claims(raw_product.description)
    }
    return cleaned
```

Data Quality Tasks:

- **Image Processing**: Resize, crop, enhance packaging images
- **Text Normalization**: Standardize ingredient lists and claims
- **UPC Validation**: Ensure proper format and check digit validation
- **Taxonomy Mapping**: Convert legacy categories to Nielsen IQ structure
- **Duplicate Detection**: Identify and merge duplicate products

Task 1.3: Training Data Preparation

```python
# Create training datasets for custom classification
training_data = {
    'organic_products': {
        'positive_samples': load_certified_organic_products(),
        'negative_samples': load_conventional_products(),
        'validation_set': load_manually_verified_organic()
    },
    'dietary_attributes': {
        'vegan': extract_vegan_training_data(),
        'gluten_free': extract_gf_training_data(),
        'keto_friendly': extract_keto_training_data()
    }
}
```

Phase 2: Label Insight Processing (4-6 weeks)

Task 2.1: Image-Based Classification

Computer Vision Pipeline:

```python
```

```
# Packaging analysis workflow
class PackagingAnalyzer:
    def __init__(self):
        self.logo_detector = LogoDetectionModel()
        self.text_extractor = OCREngine()
        self.claim_classifier = ClaimClassificationModel()

    def analyze_packaging(self, image_path):
        # Step 1: Logo and certification detection
        logos = self.logo_detector.detect_certifications(image_path)

        # Step 2: Text extraction from packaging
        text = self.text_extractor.extract_text(image_path)

        #
```