# Associate Data Scientist Assignment: AI-Powered Invoice Validation System

SURI TECHNOLOGIES

# Contents

Problem Statement ..... 3
Your Task ..... 3
Dataset ..... 3
Assignment Tasks ..... 7
Deliverables ..... 9
Evaluation Criteria ..... 10
Setup & Environment ..... 11
Bonus Points ..... 11
Time Expectation ..... 11

## Problem Statement

Suri Technologies is building an automated invoice processing system. Our customers receive invoices from various vendors and need to validate them against purchase orders (POs). The AI system must:

1. Extract key fields using OCR

2. Validate extracted data against database records

3. Highlight discrepancies directly on the document

4. Provide a user-friendly notification system

5. Track verification status

## Your Task

Develop a proof-of-concept system that demonstrates the core AI/ML capabilities for this solution. Focus on the data science components - particularly the validation logic and discrepancy detection.

## Dataset

1. Sample Invoice Images (Simulated)

We provide 50 synthetic invoice images with variations in:

- Layout templates (5 different formats)

- Font styles and sizes

- Scan quality (some with rotation, noise, poor lighting)

- Handwritten annotations (for some fields)

Note: Since we cannot share actual invoice images, we'll simulate this with structured data representations or advise you to take any picture from internet and process.

2. Ground Truth Data (ground_truth.json)

json


```
{
  "invoices": [
    {
      "invoice_id": "INV-2024-001",
      "expected_data": {
        "vendor_name": "Tech Solutions Inc.",
        "vendor_address": "123 Innovation Drive, San Francisco, CA 94107",
        "customer_name": "Suri Technologies",
        "customer_address": "456 Data Street, Austin, TX 78701",
        "po_number": "PO-78910",
        "invoice_date": "2024-01-15",
        "due_date": "2024-02-14",
        "total_amount": 12500.75,
        "tax_amount": 1125.07,
        "line_items": [
          {"description": "Cloud Storage Subscription", "quantity": 5, "unit_price": 2000, "total": 10000},
          {"description": "API Access License", "quantity": 1, "unit_price": 2500.75, "total": 2500.75}
        ]
      },
      "document_quality": 0.95
    }
  ]
}
```

3. OCR Output with Errors (ocr_results.json)

Simulated OCR outputs with realistic errors:

json

```
{
  "INV-2024-001": {
    "raw_text": "...",
    "structured_data": {
      "vendor_name": "Tech Solutions Inc",  // Missing period
      "vendor_address": "123 Innovation Dr, San Francisco, CA 94107",  // "Drive" abbreviated
      "customer_name": "Suri Tech",  // Truncated
      "customer_address": "456 Data Street, Austin, TX 78701",  // Correct
      "po_number": "P0-78910",  // 'O' misread as '0'
      "invoice_date": "2024-01-15",  // Correct
      "due_date": "2024-02-15",  // Wrong date (off by 1 day)
      "total_amount": 12500.75,  // Correct
      "tax_amount": 1125.70,  // Slight difference
      "line_items": [
        {"text": "Cloud Storage Subscription\nQty: 5\nPrice: $2,000.00\nTotal: $10,000.00"},
        {"text": "API Access License\nQty: 1\nPrice: $2,500.75\nTotal: $2,500.75"}
      ]
    },
    "confidence_scores": {
      "vendor_name": 0.92,
      "po_number": 0.85,
      "total_amount": 0.98
    },
    "bounding_boxes": {
```

```
      "vendor_name": {"x1": 100, "y1": 150, "x2": 300, "y2": 170},
      "po_number": {"x1": 400, "y1": 120, "x2": 500, "y2": 140}
    }
  }
}
```


4. Database Records (`database.json`)


```
json
{
  "purchase_orders": {
    "PO-78910": {
      "vendor": "Tech Solutions Inc.",
      "approved_amount": 12500.00,
      "valid_items": ["Cloud Storage Subscription", "API Access License"],
      "max_quantity": {"Cloud Storage Subscription": 10, "API Access License": 2},
      "tax_rate": 0.09
    }
  },
  "vendor_master": {
    "Tech Solutions Inc.": {
      "legal_name": "Tech Solutions Incorporated",
      "address": "123 Innovation Drive, San Francisco, CA 94107",
      "tax_id": "12-3456789"
    }
  },
  "customer_info": {
    "Suri Technologies": {
      "legal_name": "Suri Technologies LLC",
```

```
      "billing_address": "456 Data Street, Austin, TX 78701"
    }
  }
}
```


## Assignment Tasks

Part 1: Data Analysis & Understanding (45 mins)

1. Analyze the provided datasets

2. Identify potential challenges in:

   - Field matching (name variations, abbreviations)

   - Data type validation (dates, amounts)

   - Confidence score interpretation

3. Propose metrics to evaluate OCR accuracy

Part 2: Validation Logic Design (60 mins)

Design and implement validation functions that:

1. Compare extracted fields with ground truth using:

   - Exact matching for critical fields (PO numbers, amounts)

   - Fuzzy matching for names and addresses

   - Date validation within acceptable ranges

2. Implement discrepancy scoring:

   - Critical errors (PO mismatch, wrong vendor)

   - Warning-level discrepancies (date off by 1-3 days)

   - Informational mismatches (abbreviations, formatting)

3. Create a rule-based validation system that:

```
python
def validate_invoice(ocr_data, ground_truth, database):
    Returns: {
        'status': 'approved'|'rejected'|'needs_review',
        'discrepancies': list,
        'confidence_score': float,
        'validation_details': dict
    }
```

Part 3: ML/Data Science Components (75 mins)

1. Fuzzy matching algorithm for text fields:
   - Implement/use Levenshtein distance or similar
   - Account for common OCR errors (O/0, I/1/l)
   - Handle abbreviations and company suffixes
2. Anomaly detection for line items:
   - Detect quantity or price outliers
   - Flag items not in approved PO list
   - Calculate expected vs actual totals
3. Confidence calibration:
   - Combine OCR confidence with validation results
   - Weight different field validations appropriately
   - Provide overall document confidence score

Part 4: Output Generation (30 mins)

1. Design a discrepancy notification format:

```
json
{
  "field": "po_number",
  "issue_type": "critical",
```

```
  "expected": "PO-78910",
  "detected": "P0-78910",
  "confidence": 0.85,
  "suggestion": "Check character '0' - should be letter 'O'",
  "bounding_box": {"x1": 400, "y1": 120, "x2": 500, "y2": 140}
}
```

2. Create visualization logic (pseudocode):

```
python
def generate_visual_feedback(image, discrepancies):
    # Return coordinates and labels for rectangles
    # Highlight areas with discrepancies
```

3. Design status tracking system:

- Document verification workflow
- User correction logging
- Final approval/rejection logic

Part 5: Testing & Evaluation (30 mins)

1. Create test cases for:

- Edge cases (missing fields, poor scans)
- Common OCR errors
- Business rule violations

2. Evaluate your system on provided samples

3. Suggest improvements for production deployment

## Deliverables

1. Jupyter Notebook/Python Script containing:

- Data analysis and insights

- Implemented validation functions
- ML components for matching and anomaly detection
- Test results and performance metrics

2. Design Document (PDF or Markdown) covering:
   - Approach and methodology
   - Assumptions and limitations
   - Scalability considerations
   - Recommendations for improvement

3. Sample Outputs for 3 test invoices:
   - Validation results
   - Discrepancy notifications
   - Visualization coordinates

## Evaluation Criteria

| Category | Weight | What We're Looking For |
|---|---|---|
| Problem Understanding | 20% | Clear analysis of challenges and constraints |
| Solution Design | 25% | Logical, scalable approach to validation |
| Implementation Quality | 30% | Clean, efficient, well-documented code |
| ML/Data Science Application | 15% | Appropriate use of algorithms and techniques |
| Communication | 10% | Clear explanations and documentation |

## Setup & Environment

- Python 3.8+
- Libraries you might need: pandas, numpy, scikit-learn, fuzzywuzzy, matplotlib
- Feel free to use any additional libraries
- Provide requirements.txt if needed

## Bonus Points

1. Implement a simple ML model to predict OCR confidence based on field characteristics
2. Design a feedback loop to improve validation rules
3. Propose a data collection strategy for model improvement
4. Consider multilingual invoice handling
5. Suggest privacy-preserving techniques for sensitive data

## Time Expectation

- Total: 16 hours (2 business days)
- You may prioritize sections based on your strengths
- Focus on demonstrating your data science thinking process

Good luck! We're excited to see your approach to this real-world problem.

Note: This is a simulated assignment. All data is synthetic and created for evaluation purposes only.