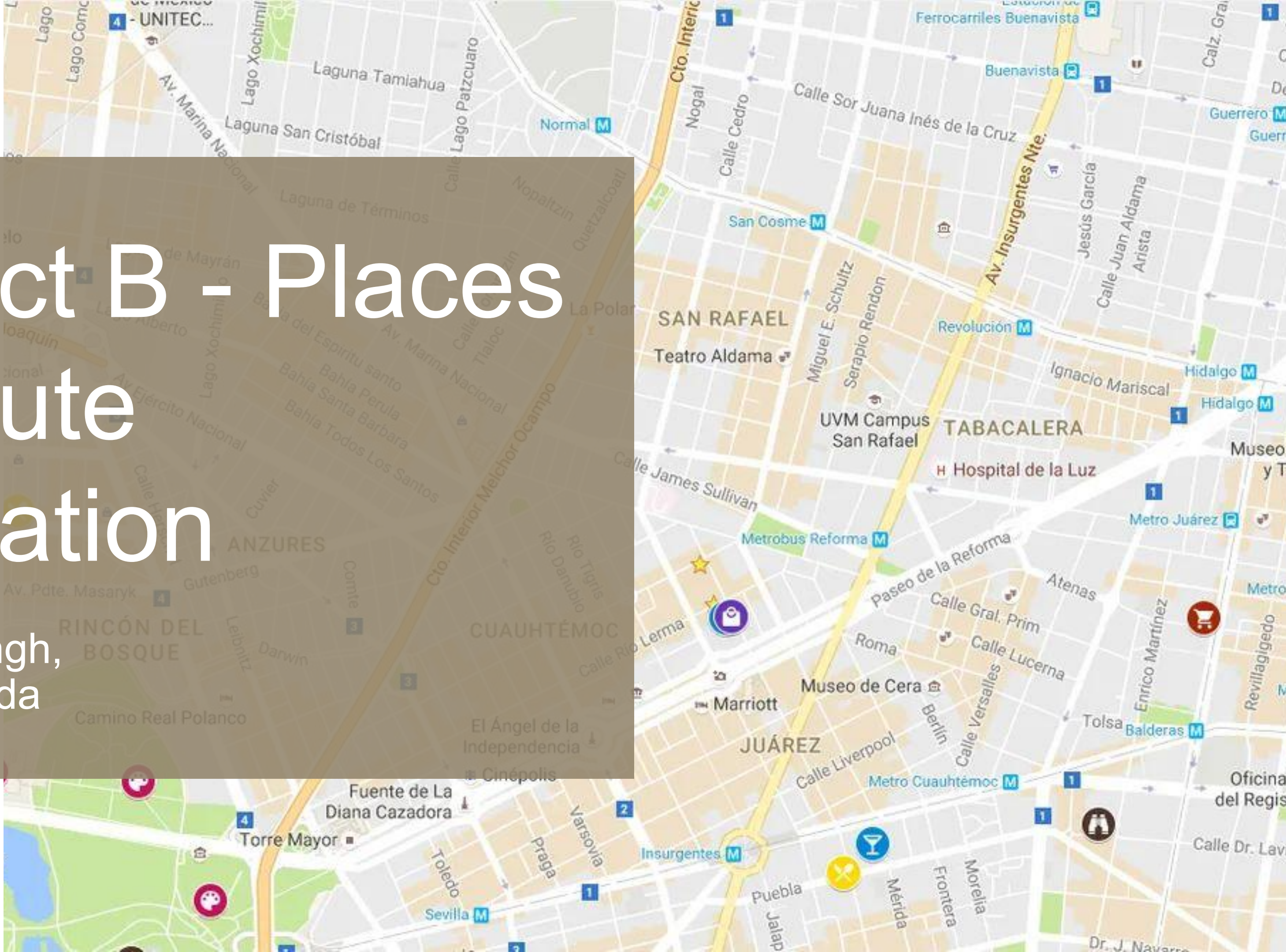
Project B - Places Attribute Conflation

Jaskaran Singh,
Varnit Balivada



Problem Overview:

- Real-world place → many data sources → conflicting attributes
  - EX: name variants, outdated phone numbers, different websites

- Need: one clean, unified record per place

- Project question: “When sources disagree, how do we automatically choose the best attribute; rules, ML, both?

Attribute Conflation is Messy


```
--- Sample Conflicting Record ---
Record ID: 08f44f055a9a016e0390f050aa3c93c0

Current:
  Name: Goin' Postal Jacksonville
  Phone: ["+19049989600"]
  Address: [{"freeform":"7643 Gate Pkwy","locality":"Jacksonville","postcode":"32256-2892","region":"FL","country":"US"}]

Base:
  Name: Goin' Postal Jacksonville
  Phone: ["9049989600"]
  Address: [{"freeform":"7643 Gate Pkwy Ste 104","locality":"Jacksonville","region":"FL","country":"US","postcode":"32256"}]
```

Original Objectives & Key Results

Objective 1 – Ground Truth Dataset

KR1: 5,000 labeled “Golden Dataset” records

KR2: >95% inter-annotator agreement on 200 records

KR3: Define 5 key attributes + guidelines

Objective 2 – Superior Selection Algorithm

KR1: Beat “most recent” heuristic by ≥15% F1

KR2: F1 > 0.90 on name attribute (1,000 places)

KR3: Resolve >99% of places automatically

Objective 3 – Recommendation for Overture

KR1: Comparative ML vs rule-based report

KR2: Identify top 3 edge cases

KR3: Cost–benefit + determine if ML > Human algorithm

How Our OKRs Evolved

Objective 1 – Ground Truth Dataset:

KR1: 2,000 labeled “Golden Dataset” records

KR2: ≥80% inter-annotator agreement on 200 records + ≥10 disagreement patterns

KR3: 5 key attributes w/ ≥15 edge cases

Objective 2 – Compare Selection Approaches

KR1: 3 approaches (rule-based, ML, hybrid) with ≥60% accuracy

KR2: ≥20 failure cases per approach

KR3: Inference <100ms per record within practical memory limits

Objective 3 – Recommendation for Overture

KR1: Technical report + comparison across ≥5 metrics

KR2: Identify top 5 edge cases

KR3: 2 system designs (high-volume/low-latency & high-accuracy/manual-review)

KR4: Comparison matrix + recs for ≥2 use cases

OKR Pivot

- 5,000 record “Golden Dataset” → 2,000 normalized Yelp, 200 manual annotated Overture Sample Set
- Focus multiple approaches rather than a singular “superior” one
    - Logging inference time / memory for future scalability
- Document guidelines, edge cases, comparison across metrics for future development

Goals Met

Obj 1 (Data): 2,000 Synthetic Records; 100% IAA Validation Set

```
Results:
────────────────────────────────────────────────────────────
Metric                       | Value
─────────────────────────────|──────────────────
AI Agreement (Qwen/Gemma)    | 46% (92/200 agreed)
Disagreements                 | 108 records
Cohen's Kappa                 | 0.227 (Fair)
Manual Review                 | 200/200 resolved (100%)
Final Validation Set          | 200 records (diamond standard)
────────────────────────────────────────────────────────────

Disagreement Patterns Documented:
  ✓ Base vs Current conflicts (79 cases)
  ✓ Capitalization differences
  ✓ Completeness vs Formatting trade-offs
  ✓ Category structure variations
  ✓ URL quality differences
  [Total: ≥10 distinct patterns]

Status: ✅ ACHIEVED (100% resolution after manual review)
```

Obj 2 (Models): 3 Approaches Built (ML, Rules, Hybrid) w/ ≥60% accuracy; 0.002ms Latency

```
### Compute / Scalability Metrics (KR3)
_Logged via `scripts/run_inference.py` + `docs/OKRs.md`_

| Metric | Value |
|--------|-------|
| Inference speed (per record) | **~0.0066 ms** on `project_b_samples_2k.parquet` |
| Peak memory usage | **≈200 MB** |
| Hardware | Local CPU (no GPU required for inference) |
| Pipeline | End-to-end run via `scripts/run_algorithm_pipeline.py` |

These figures are well under the <100 ms/record target for 150–200M places/month.
```

Obj 3 (Insights): Failure Analysis & Design Proposals Delivered

```
### ML Model Failures (Count: 108)
**1. Record 08f2986b8089b2cb03220db3aa72c816**
- Prediction: BASE | Truth: CURRENT
- Current: `{'freeform': '3663 S Las Vegas Blvd', 'locality': 'Las Vegas', 'postcode': '89109', 'region': 'NV', 'country': 'US'}`
- Base:    `{'freeform': '3663 Las Vegas Blvd South', 'locality': 'Las Vegas', 'region': 'NV', 'country': 'US', 'postcode': '89109'}`
```

```
### Baseline (Most Recent) Failures (Count: 77)
**1. Record 08f2aa859dba3af003731e8ef72ef347**
- Prediction: SAME | Truth: CURRENT
- Current: `["https://international-bakery-inc-md.hub.biz"]`
- Base:    `None`
```

```
### ML Model Failures (Count: 124)
**1. Record 08f3da18ccad52ad03cc06b87820910f**
- Prediction: BASE | Truth: CURRENT
- Current:  `davaindia GENERIC PHARMACY`
- Base:     `Davaindia Generic Pharmacy`
```

Establishing Ground Truth: Golden Dataset


```
{
  "id": "synthetic_yelp_c4Y_RZKBXsXENA9y7JIBaQ",
  "data": {
    "current": {
      "names": "{\"primary\": \"kool tortas\"}",
      "phones": "[\"(599) 376-7457\"]",
      "websites": "[\"http://kooltortas.com\"]",
      "addresses": "[{\"freeform\": \"4547 S 6th Ave\", \"locality\": \"Tucson\",
      \"region\": \"AZ\", \"country\": \"US\"}]",
      "categories": "{\"primary\": \"Mexican\", \"alternate\": []}"
    },
    "base": {
      "names": "{\"primary\": \"Kool Tortas\"}",
      "phones": "[\"+15993767457\"]",
      "websites": "[\"https://www.kooltortas.com\"]",
      "addresses": "[{\"freeform\": \"4547 S 6th Ave\", \"locality\": \"Tucson\",
      \"region\": \"AZ\", \"postcode\": \"85714\", \"country\": \"US\"}]",
      "categories": "{\"primary\": \"Mexican, Restaurants\"}"
    }
  },
  "label": "b",
  "method": "synthetic_yelp_proxy"
}
```


```
**Key Differences** (simulated noise):
- **Name**: `"kool tortas"` (lowercase) vs `"Kool Tortas"` (proper case)
- **Phone**: `"(599) 376-7457"` vs `"+15993767457"` (formatting difference)
- **Website**: `"http://kooltortas.com"` vs `"https://www.kooltortas.com"` (HTTP vs HTTPS,
www)
- **Address**: Missing postcode in current vs complete in base
- **Category**: Simple `"Mexican"` vs detailed `"Mexican, Restaurants"`
- `label: "b"` = Base is better (more complete + proper formatting)
```


```
{
  "id": "08f64a59990a63a0030784450be43ba4",
  "record_index": 0,
  "label": "b",
  "method": "manual_review (manual)",
  "data": {
    "current": {
      "names": "{\"primary\":\"ร้านตัดผมพี่ริน ห้วยตะแคง\"}",
      "phones": NaN,
      "websites": NaN,
      "addresses": "[{\"country\":\"TH\"}]",
      "categories": "{\"primary\":\"beauty_salon\",\"alternate\":[\"barber\",\"thai_restaurant\"]}",
      "confidence": 0.2315484804630969
    },
    "base": {
      "names": "{\"primary\":\"ร้านตัดผมพี่ริน ห้วยตะแคง\",\"common\":{},\"rules\":[]}",
      "phones": "[null]",
      "websites": "[null]",
      "addresses": "[{\"country\":\"TH\"}]",
      "categories": "{\"primary\":\"Business and Professional Services > Health and Beauty Service > Hair Salon\",\"alternate\":[]}",
      "confidence": 1.0
    }
  }
}
```


```
**Key Fields**:
- `label: "b"` = **Base** version is better (more structured category)
- Shows real-world edge cases: missing data, international characters, category hierarchy differences
```

Data sources:

- 2,000-record Synthetic Golden Dataset generated from Yelp businesses = used for training.
- 2,000 pre-matched Overture places for final inference + evaluation
- 200 Overture records fully human-validated from AI + manual review

Key attributes:

- Name, phone, website, address, category
- Detailed guidelines + edge cases (formatting, abbreviations, partial matches, etc.)

Annotation pipeline:

- annotate_ai.py – connects to local LLMs (Qwen, Gemma)
- Auto-save, resume, progress tracking
- review_disagreements.py for manual review of conflicting labels

Two datasets we ended up with:

- Synthetic 2k Yelp dataset → primary training/evaluation for experimentation
- Manual 200 Overture dataset → ‘diamond standard’ for real-world validation and failure analysis

Our Pipeline

Raw Data

```
ID: 08f44f055a9a016e0390f050aa3c93c0

Current Attributes:
  • Name: {"primary":"Goin' Postal Jacksonville"}
  • Confidence: 0.9963
  • Sources: [multiple data sources]

Base Attributes:
  • Name: {"primary":"Goin' Postal Jacksonville"}
  • Confidence: 0.77
  • Sources: [base dataset]

Conflict: Same name, different confidence scores
Task: Choose best attribute (current vs base)
```

Golden Dataset


```
{
  "id": "08f4823ad167532903b45bc53a70221e",
  "record_index": 2,
  "label": "b",
  "method": "manual_review (manual)",
  "data": {
    "current": {
      "names": "{\"primary\":\"The Home Depot México\"}",
      "phones": "[\"+528000046633\"]",
      "websites": "[\"http://www.homedepot.com.mx/\"]",
      "addresses": "[{\"freeform\":\"Calle Benito Juárez 75\",\"locality\":\"Ca
      "categories": "{\"primary\":\"home_and_garden\",\"alternate\":[\"hardware
      "confidence": 0.7927927927927928
    },
    "base": {
      "names": "{\"primary\":\"The Home Depot\",\"common\":{},\"rules\":[]}",
      "phones": "[null]",
      "websites": "[\"http://www.homedepot.com\"]",
      "addresses": "[{\"freeform\":\"Carretera Transpeninsular Benito Juarez Km
      "categories": "{\"primary\":\"Retail > Hardware Store\",\"alternate\":[\"
      "confidence": 1.0
    }
  }
},
```


Feature Extraction

```
Metadata Features:
  • confidence_current:       1.0
  • confidence_base:          0.5
  • confidence_diff:         0.5
  • confidence_ratio:        2.0
  • sources_current_count:   0.0
  • sources_base_count:      0.0

String Similarity Features:
  • name_exact_match:        0.0
  • name_exact_match_lower:  1.0
  • name_length_ratio:        1.0
  • name_levenshtein_similarity: 1.0
  • name_jaro_winkler_similarity: 1.0

Formatting Features:
  • name_current_length:      [length]
  • name_base_length:         [length]
  • name_length_diff:          [difference]
  • [capitalization flags, punctuation, etc.]
```

Training

```
Best Model: Logistic Regression

Metrics:
  • Accuracy:  98.4%
  • Precision: 98.4%
  • Recall:    99.5%
  • F1-Score:  98.9% ✓ (exceeds KR2 target of 90%)
  • Coverage:  100%  ✓ (exceeds KR3 target of 99%)
```

Evaluation


```
"algorithm": "ML Model (category)",
"metrics": {
  "accuracy": 0.715,
  "precision": 0.715,
  "recall": 1.0,
  "f1": 0.8338192419825073,
  "n_samples": 200
},
"coverage": 1.0,
"n_total": 200,
"n_unclear": 0
```

```
"algorithm": "ML Model (phone)",
"metrics": {
  "accuracy": 0.72,
  "precision": 0.7351351351351352,
  "recall": 0.951048951048951,
  "f1": 0.8292682926829268,
  "n_samples": 200
},
"coverage": 1.0,
"n_total": 200,
"n_unclear": 0
```


Inference

```
Record ID: 08f44f055a9a016e0390f050aa3c93c0

Current Name: {"primary":"Goin' Postal Jacksonville"}
Base Name:    {"primary":"Goin' Postal Jacksonville"}

Predictions:
  • Most Recent Rule:    CURRENT (always picks current)
  • Confidence Rule:     SAME (confidence diff < threshold)
  • Completeness Rule:   CURRENT (tie-breaker)
  • Hybrid Ensemble:     SAME (weighted voting)
  • ML Model:            [would output probability]

Final Decision: SAME (names are identical)

Performance Metrics:
  • Inference Time: ~0.0066 ms per record
  • Memory Usage: ~200 MB
  • Scalability: Handles 150-200M places/month ✓
```

Selection Approaches: Rule-Based

Rule-Based

- Most Recent Rule:
  - If current & base → current
  - If one is missing → pick the other
  - Identical → “same”
- Confidence Rule (confidence vs base_confidence):
  - Exceeds threshold → pick that
  - Close call → “same”
- Completeness Rule:
  - Higher completeness (extra fields, non-empty, etc.) → picked

Real record from `data/project_b_samples_2k.parquet`

```text
Record ID: 1407374885933937

Current name: {"primary":"Red Wing - Roswell, GA"}
Base name:    {"primary":"Red Wing"}

Rule logic:
1. Both exist? ✓ Yes
2. Are they identical? ✗ No ("Red Wing - Roswell, GA" ≠ "Red Wing")
3. Decision: Always pick current (most recent assumption)

Most Recent Rule prediction: CURRENT
```

Most Recent Rule

Real record from `data/project_b_samples_2k.parquet`

```text
Record ID: 08f3956260b9e14003feca2bf0764d0c

Current name: {"primary":"Norauto España"}
Base name:    {"primary":"Norauto"}

Confidence scores:
- current_confidence = 0.9963
- base_confidence    = 0.7700
- difference         = 0.2263

Rule logic:
1. Both exist? ✓ Yes
2. Are they identical? ✗ No
3. Confidence difference (0.2263) > threshold (0.05)? ✓ Yes
4. Current confidence is higher? ✓ Yes

Confidence Rule prediction: CURRENT
```

Confidence Rule

Real record from `data/project_b_samples_2k.parquet`

```text
Record ID: 1407374885933937

Current name: {"primary":"Red Wing - Roswell, GA"}
Base name:    {"primary":"Red Wing"}

Completeness calculation:
- Current: Has "primary" field ✓ (score +0.5), only 1 field (score +0.0) = 1.5
- Base:    Has "primary" field ✓ (score +0.5), only 1 field (score +0.0) = 1.5

Rule logic:
1. Both exist? ✓ Yes
2. Are they identical? ✗ No
3. Completeness scores equal? ✓ Yes (both 1.5)
4. Tie-breaker: Default to current (recency)

Completeness Rule prediction: CURRENT
```

Completeness Rule

Selection Approaches: ML

ML

- (current, base) → numeric features = model learns patterns
- Feature extraction:
  - String similarity
  - Formatting
  - Metadata
- Models:
  - Logistic Regression
  - Random Forest
  - Gradient Boosting
- Training
  - Input: feature vector + label
  - Output: probability current > base
- Prediction:
  - Prob > 0.5 → "current"
  - Else → "base"

```python
def extract_features_for_record(row: pd.Series, attribute: str = 'name') -> Dict[str, float]:
    features = {}
    metadata_features = extract_metadata_features(row)
    features.update(metadata_features)

    if attribute == 'name':
        name_features = extract_name_features(row['names'], row['base_names'])
        features.update(name_features)

    return features
```

Figure 1

Extracted via `extract_features_for_record` *on the first record of* `project_b_samples_2k.parquet`

```
{
  "confidence_current": 0.9963,
  "confidence_base": 0.7700,
  "confidence_diff": 0.2263,
  "confidence_ratio": 1.2938,
  "sources_current_count": 24,
  "sources_base_count": 4,
  "sources_count_diff": 20,
  "name_exact_match": 1.0,
  "name_exact_match_lower": 1.0,
  "name_length_ratio": 1.0,
  "name_levenshtein_similarity": 1.0,
  "name_jaro_winkler_similarity": 1.0
  // ... ~15 more name-specific features (capitalization flags, punctuation, etc.)
}
```

Figure 2

```json
{
  "best_model": "logistic_regression",
  "best_val_f1": 0.9900332225913622,
  "best_val_acc": 0.985,
  "all_models": {
    "logistic_regression": {
      "val_f1": 0.9900332225913622,
      "val_acc": 0.985,
      "model_type": "logistic_regression"
    },
    "random_forest": {
      "val_f1": 0.9797979797979798,
      "val_acc": 0.97,
      "model_type": "random_forest"
    },
    "gradient_boosting": {
      "val_f1": 0.9865771812080537,
      "val_acc": 0.98,
      "model_type": "gradient_boosting"
    }
  }
}
```

Figure 3

Selection Approaches: Hybrid

Hybrid

- Weighted voting across 3 rules
- How it works:
  - Each rule → "current", "base", "same" given weight
  - Each record = add weight depending on rule output to:
    - score_current
    - score_base
    - Score_same
  - Highest score = final prediction
- All 3 agree → Model follows
- Disagree → higher-weighted rules win

```python
class HybridBaseline:
    def __init__(self, recency_weight: float = 0.3,
                 confidence_weight: float = 0.5,
                 completeness_weight: float = 0.2):
```

Extracted from `data/project_b_samples_2k.parquet` using `baseline_heuristics.py`

```
Record ID: 1407374885933937
Current name: {"primary":"Red Wing - Roswell, GA"}
Base name:    {"primary":"Red Wing"}

Individual rule outputs (name selector):
- MostRecentBaseline     → current
- ConfidenceBaseline     → same
- CompletenessBaseline   → current

Hybrid weights:
- recency_weight      = 0.3
- confidence_weight   = 0.5
- completeness_weight = 0.2

Hybrid score accumulation:
- From MostRecentBaseline (current):
    score_current += 0.3
- From ConfidenceBaseline (same):
    score_same    += 0.5
- From CompletenessBaseline (current):
    score_current += 0.2

Final scores:
- score_current = 0.3 + 0.2 = 0.5
- score_same    = 0.5
- score_base    = 0.0

Hybrid prediction: SAME (ties current in total weight, but confidence vote pushes it toward equivalence)
```

Rule-Based, ML, and Hybrid

Approach	Where it Wins	Strengths	Weaknesses
Rule-Based	Name, Phone, Website	• Deterministic • Zero training • <1ms Inference	• Misses nuanced quality signals • Brittle when noisy data
ML	Category (ties Rules)	• Captures subtle patterns • Synthetic Data -> High accuracy	• Label quality dependant • Name = low F1 score
Hybrid	Address, Phone (ties Rules)	• Balance of accuracy + interpretability • Obvious cases = Rules handle	• Inherits rule's limitations • Needs tuning per attribute

Performance Metrics (F1-Score on 200 Real-World Records):

Attribute	Best Approach	F1-Score	ML F1	Baseline F1	Hybrid F1
Category	ML / Hybrid	0.8338	0.8338	0.8338	0.8094
Address	Hybrid / Rules	0.8338	0.7921	0.8338	0.8338
Phone	Hybrid / Rules	0.8554	0.6929	0.8554	0.8554
Website	Hybrid / Rules	0.8323	0.4600	0.8323	0.8323
Name	Rule-Based	0.8338	0.2209	0.8338	0.7667

```
_From `docs/OKRs.md` and `scripts/run_inference.py`_

| Approach        | Training Time              | Inference Time (per record) | Memory Usage |
|-----------------|----------------------------|-----------------------------|--------------|
| **Rule-Based** | 0 seconds (no training)    | **~0.001 ms**               | <10 MB       |
| **ML**          | ~30-60 seconds (one-time) | **~0.0066 ms**              | ~200 MB      |
| **Hybrid**      | 0 seconds (no training)    | **~0.002 ms**               | <10 MB       |
```

Results: How Well Do the Approaches Work?

Quantitative (on current datasets)

- Synthetic 2K (Yelp) – Name attribute
  - Logistic Regression: F1 ≈ 0.99, Accuracy ≈ 98%
  - Most Recent baseline: F1 ≈ 0.75, Accuracy ≈ 75%
- Real 200 (human-labeled Overture) – per attribute
  - Name: ML struggles (F1 ~0.22) vs rules (F1 ≈ 0.83)
  - Phone / Category: ML ≈ 0.80–0.83, competitive with rules
  - Website: ML ≈ 0.73, rules often stronger
  - Address: ML struggles; rule-based methods outperform

Qualitative

- Low AI agreement (46%) exposed ambiguity in attribute selection
- Disagreement cases = better guidelines and model thresholds

For Overture

- Rules often win on real data (Name/Phone/Website)
- All approaches meet speed requirements
- Deploy rules as primary; hybrid as alternative where it matches

```
### Synthetic 2K (Yelp) – Name Attribute Test Set
| Selector | Accuracy | Precision | Recall | F1 | Notes |
|----------|----------|-----------|--------|----|-------|
| Logistic Regression | **0.9840** | 0.9842 | 0.9947 | **0.9894** | 1,000-record held-out test |
| Most Recent Baseline | 0.7510 | – | – | ≈0.75 | Always picks current version |
| Coverage | 100% for both | | | | No “unclear” predictions |
```

```
### Real 200 (Manual Overture) – Per Attribute
| Attribute | Selector | Accuracy | Precision | Recall | F1 | Coverage |
|-----------|----------|----------|-----------|--------|----|----------|
| **Name** (183 usable) | Logistic Regression | 0.5683 | 0.6621 | 0.7619 | **0.7085** | 100% |
| Name Baseline (Most Recent)
| **Phone** | Logistic Regression | 0.7200 | 0.7351 | 0.9510 | **0.8293** | 100% |
| Phone Baseline (Most Recent)
| **Website** | Logistic Regression | 0.6200 | 0.7410 | 0.7203 | **0.7305** | 100% |
| **Address** | Logistic Regression | 0.3750 | 0.8750 | 0.1469 | 0.2515 | 100% |
| **Category** | Logistic Regression | 0.7150 | 0.7150 | 1.0000 | **0.8338** | 100% |
```

```
### Real 200 – ML vs Rule-Based (per attribute)

| Attribute | Best Rule F1 (Most Recent / Completeness / Hybrid) | ML F1 | Who Wins? |
|-----------|-----------------------------------------------------|-------|----------|
| Name      | ≈ **0.83** (rules)                                  | ~0.22 | Rules by a lot |
| Phone     | ≈ **0.86** (rules)                                  | 0.83  | Close, rules slightly better |
| Website   | ≈ **0.83** (rules)                                  | 0.73  | Rules better |
| Address   | ≈ **0.83** (rules)                                  | 0.25–0.80* | Rules better overall |
| Category  | ≈ **0.83** (rules)                                  | 0.83  | Rough tie |
```

Error Analysis & Real-World Constraints

Error taxonomy: ≥5 categories

- Slight formatting differences ("St" vs "Street")
- Old vs new names / rebrands
- Partial addresses / missing unit numbers
- Suspicious websites vs Yelp URLs
- Category mismatch (restaurant vs café vs bar)

```
**4. Record 08f446c25679a70e03572240a924ba2c**
- Prediction: BASE | Truth: CURRENT
- Current: `Chick-fil-A Grand Parkway North`
- Base:    `Chick-fil-A`
```

Production Realities

- Clusters of 10–100 places, not just pairs

- Can conflate 150–200M places/month → inference time matters

Our response

- Benchmark compute requirements
- Propose 2 designs:
  - High-volume/low-latency pipeline
  - High-accuracy pipeline with manual review for high-risk cases

Live Demo:

[GitHub]

Reflection & Future Work

What We'd Do Differently with Another Quarter

- Start with edge-case taxonomy earlier
- Acquire more data
- Mix human annotation earlier w/ model-generated labels
- Research further into model annotation for decision making

Open Questions / Future Work

- Begin implementing the future design proposals into our model
- Create a single, unified hybrid model
- Exploring weights (i.e Websites & Phone for Restaurants)
- What should trigger manual review, when can a model NOT make an acceptable decision?

Team Growth & Learnings

- Iterative OKR refinement > constant planning
- Sponsor feedback improved direction
- Tooling investment (annotation + disagreement review) saved huge manual effort
- Industry communication can be slow, solve blockers quick

THANK YOU, QUESTIONS?

Contact:
Varnit Balivada
vbalivad@ucsc.edu

Jaskaran Singh
jask@ucsc.edu

GitHub