Indiana Department of Health

Probabilistic Record Linkage with Splink

Matt Simmons
Office of Data & Analytics

5/29/2024

OUR MISSION:

To promote, protect, and improve the health and safety of all Hoosiers.

OUR VISION:

Every Hoosier reaches optimal health regardless of where they live, learn, work, or play.

Outline

Goals – record linkage techniques, probabilistic math, splink demo

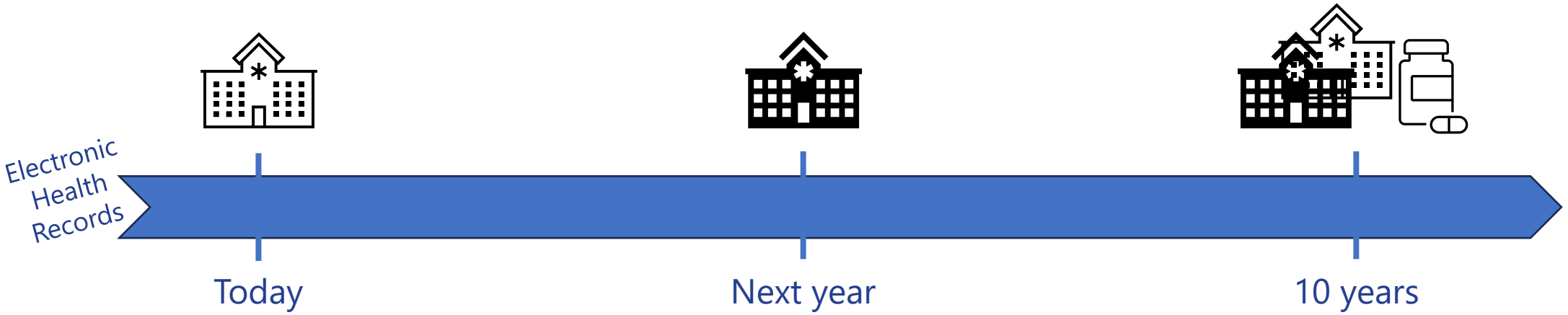
1. Motivation
2. Entity Resolution Techniques
3. splink
4. Validation
5. Next Steps
6. Questions/Discussion

Motivation

Indiana Department of Health

Why does this matter?

- Clinical and public health continuity of care
- Holistic patient view and reduction of bias
- Novel insights
- IDOH – COVID cases and vaccinations



Indiana Department of Health

Patient Entity Deduplication

In the simplest use case of patient entity data, how would this dataset be de-duplicated?

MRN	Name	DOB	City	Hospital	Admit Date
1000	John Smith	1/4/1980	Indianapolis	A	9/7/2018
1001	Smith, Jon	1/4/1980	Indianapolis	A	2/20/2021
1001	Jonny A Smith	4/1/1980	Avon	B	5/11/2024
1002	Smith, Johnathan	1-Jan-80	Indianapolis	A	6/19/2022

3

3

4…?

Availability of Identifiers

From Intermountain's 6.6M record Master Patient Index (MPI) database

Sequence	Combination of Traits	Completeness	Uniqueness
1	FN+LN+DoB	98.2%	95.7%
2	FN+LN+DoB+Sex	98.2%	95.9%
3	FN+LN+DoB+Sex+ZIP(first 5)	91.1%	99.2%
4	FN+LN+DoB+Sex+Phone	76.2%	99.5%
5	FN+LN+DoB+Sex+MN	59.9%	98.9%
6	FN+LN+DoB+Sex+MN(initial)	60.0%	97.7%
7	FN+LN+DoB+Sex+SSN(last 4)	61.9%	99.7%

Attribute Name	Completeness	Validity	Distinctiveness	Comparability	Stability
EMPI	100%	--	100%	Very High	Very High
Last Name	99.85%	99.84%	5.1%	Medium	High
First Name	99.85%	99.33%	3.1%	Medium	High
Middle Name	60.54%	60.54%	2.6%	Medium	High
Suffix Name	0.08%	0.08%	0.08%	Medium	Medium
SSN	61.40%	60.92%	98.0%	High	High
Sex (Admin. Gender)	99.98%	99.98	0.00008%	High	High
Date of Birth	98.18%	97.38%	0.8%	High	Very High
Date of Death	3.36%	3.36%	3.4%	High	Very High
Street Address (1 or 2)	95.00%	94.61%	44.4%	Low	Low
City	94.84%	94.83%	0.8%	High	Low
State	94.81%	94.39%	0.8%	High	Low
Facility MRN	99.90%	99.90%	99.90%	High	Low
Postal Code	92.31%	92.0%	0.6%	High	Low
Primary Phone Number	90.68%	87.26%	51.6%	High	Medium
Work Phone Number	20.28%	19.79%	51.6%	High	Low
Ethnicity	25.25%	25.25%	0.0003%	High	Very High
Race	76.25%	76.25%	0.0001%	High	Very High

https://sequoiaproject.org/wp-content/uploads/2015/11/The-Sequoia-Project-Framework-for-Patient-Identity-Management.pdf

Entity Resolution

Indiana Department of Health

Deduplication/Linking Techniques

Deterministic – an exact key-value matching using one or more fields

- MRN == MRN
- (MRN == MRN) & (Name == Name) & (DOB == DOB)
- concat(Name + DOB) == concat(Name + DOB)

MRN	Name	DOB	City	Hospital
1000	John Smith	1/4/1980	Indianapolis	A
1001	Smith, Jon	1/4/1980	Indianapolis	A
1001	Jonny A Smith	4/1/1980	Avon	B
1002	Smith, Johnathan	1-Jan-80	Indianapolis	A

MRN	Name	DOB	City	Pharmacy
1000	John Smith	1/4/1980	Indianapolis	X
1001	Smith, Jon	1/4/1980	Indianapolis	Y
1001	Jonny A Smith	4/1/1980	Avon	Z
1002	Smith, Johnathan	1-Jan-80	Indianapolis	Z

Deduplication/Linking Techniques

Fuzzy – measures the number of primitive edits needed to make one string match another, and uses that score to approximate the probability of a match

- e.g. spell check, Google searches
- Jon → Jonny = Jon n y → 2 additions

First Name	Last Name	DOB	City	Hospit
John	Smith	1/4/1980	Indianapolis	A
Smith	Jon	1/4/1980	Indianapolis	A
Jonny	Smith	4/1/1980	Avon	B
Smith	Johnathan	1-Jan-80	Indianapolis	A

First Name	Last Name	DOB	City	Pharmac
John	Smith	1/4/1980	Indianapolis	X
Smith	Jon	1/4/1980	Indianapolis	Y
Jonny	Smith	4/1/1980	Avon	Z
Smith	Johnathan	1-Jan-80	Indianapolis	Z

String Comparison Algorithms

Method Name	Description
Levenshtein distance	counts the number of deletions, insertions and substitutions necessary to turn b into a
Optimal string alignment	is like the Levenshtein distance but also allows transposition of adjacent characters (only once, though)
Full Damerau-Levenshtein distance	is like the optimal string alignment distance except that it allows for multiple edits on substrings
Jaro distance	$sim_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{\lvert s_1\rvert} + \frac{m}{\lvert s_2\rvert} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases}$ • $\lvert s_i\rvert$ is the length of the string s_i; • m is the number of *matching characters* • t is the number of *transpositions*
Jaro-Winkler distance	adds a correction term to the Jaro-distance
Soundex	strings are translated to a soundex code based on its phonetics

Indiana Department of Health

https://journal.r-project.org/archive/2014-1/loo.pdf
https://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance

Deduplication/Linking Techniques

$$Pr(\text{Match} \mid \text{Observation}) = \frac{2^{\log_2\left(\frac{\lambda}{1-\lambda}\right)+\log_2\left(\prod_i^{\text{features}} \frac{m_i}{u_i}\right)}}{1+2^{\log_2\left(\frac{\lambda}{1-\lambda}\right)+\log_2\left(\prod_i^{\text{features}} \frac{m_i}{u_i}\right)}}$$

$$= \frac{\left(\frac{\lambda}{1-\lambda}\right)\prod_i^{\text{features}} \frac{m_i}{u_i}}{1+\left(\frac{\lambda}{1-\lambda}\right)\prod_i^{\text{features}} \frac{m_i}{u_i}}$$

$$= 1-\left[1+\left(\frac{\lambda}{1-\lambda}\right)\prod_i^{\text{features}} \frac{m_i}{u_i}\right]^{-1}$$

Probabilistic – uses a probability model to calculate match weights, which are then used to determine the probability of a match

First Name	Last Name	DOB	City	Hospit
John	Smith	1/4/1980	Indianapolis	A
Smith	Jon	1/4/1980	Indianapolis	A
Jonny	Smith	4/1/1980	Avon	B
Smith	Johnathan	1-Jan-80	Indianapolis	A

First Name	Last Name	DOB	City	Pharmac
John	Smith	1/4/1980	Indianapolis	X
Smith	Jon	1/4/1980	Indianapolis	Y
Jonny	Smith	4/1/1980	Avon	Z
Smith	Johnathan	1-Jan-80	Indianapolis	Z

https://moj-analytical-services.github.io/splink/topic_guides/theory/fellegi_sunter.html

Technique Summary

- Deterministic is not robust enough, fuzzy is too cumbersome, and probabilistic is too unapproachable
- Need a simple, portable, and scalable record linkage framework to do the best matching with the least amount of effort
- Grassroots development, enterprise software, or adopt existing library

splink

Indiana Department of Health

What is splink?

Fast, accurate and scalable probabilistic data linkage

Splink is a Python package for probabilistic record linkage (entity resolution) that allows you to deduplicate and link records from datasets that lack unique identifiers.

Key Features

⚡ Speed: Capable of linking a million records on a laptop in around a minute.

🎯 Accuracy: Support for term frequency adjustments and user-defined fuzzy matching logic.

🌐 Scalability: Execute linkage in Python (using DuckDB) or big-data backends like AWS Athena or Spark for 100+ million records.

🎓 Unsupervised Learning: No training data is required for model training.

📊 Interactive Outputs: A suite of interactive visualisations help users understand their model and diagnose problems.

Splink's linkage algorithm is based on Fellegi-Sunter's model of record linkage, with various customisations to improve accuracy.

Theory

Fellegi-Sunter model – most probabilistic RL applications are grounded in this Bayesian framework

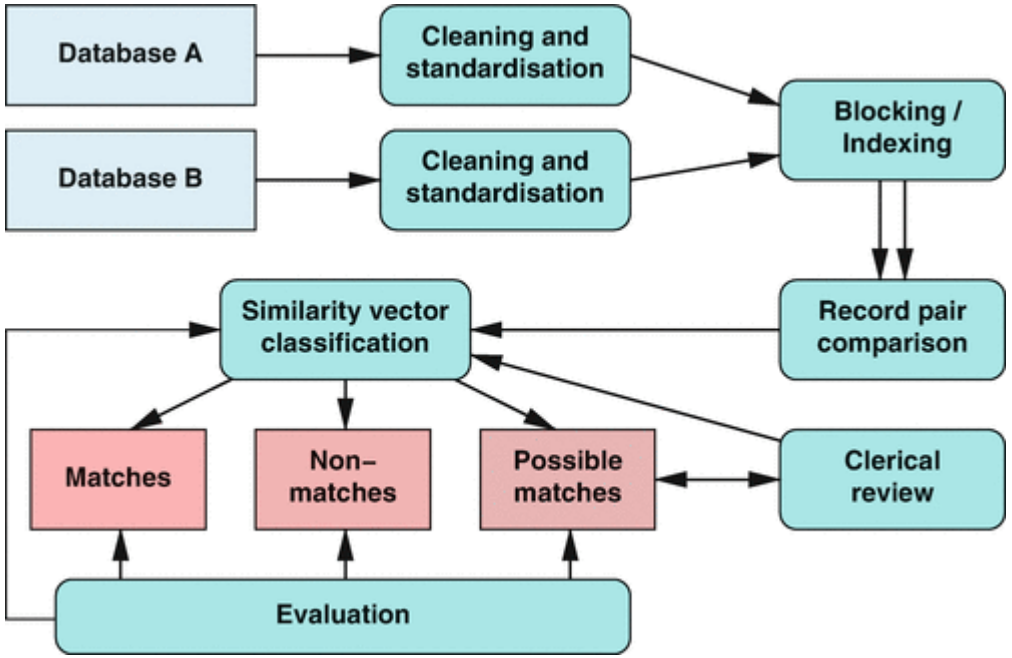
- The features of the model are the matching fields that can uniquely identify the entity and are independent from one another
- Creates a Cartesian product of the observations to compute pairwise conditional probabilities to estimate the model parameters
- Model is run with the estimated parameters, and weights of each feature are summed to calculate p for each matching pair

Workflow

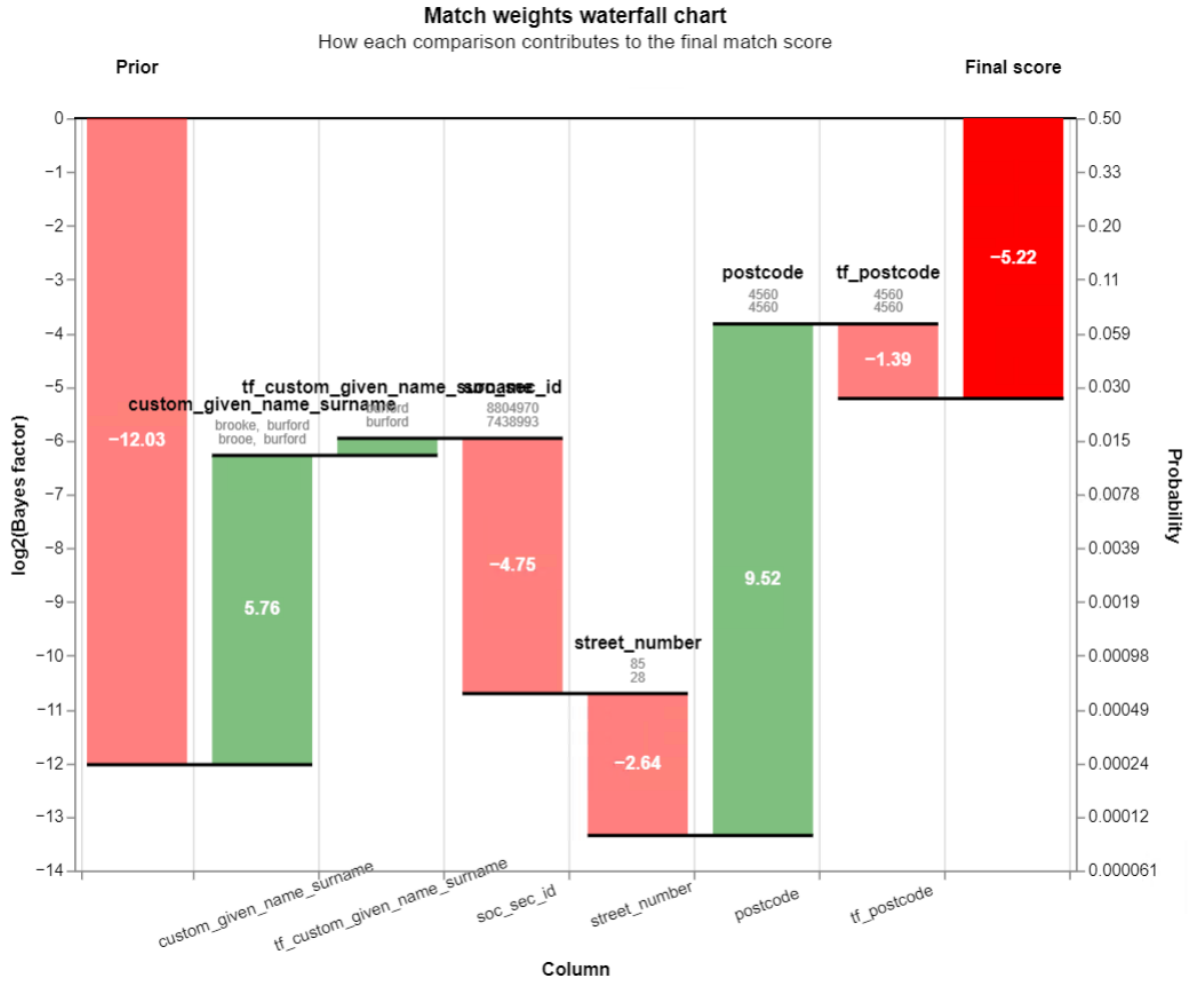
Splink combines the flexibility of **DIY** with the simplicity of pre-built object-oriented methods

- Provides many high-level methods to set up the model, but there is no UI/application element. Everything is done in Python
- Script workflow:
  1. **Import, clean and standardize** your data (esp. the model features)
  2. **Blocking rules list** (to avoid an *m x n* cardinality explosion)
     - When creating blocking rules, write them with this in mind: "only bother matching candidate pairs if one of these is true"
  3. **Settings dict** (details of how the features interact)
  4. Initialize model with backend and parameter estimation - μ, m, u
  5. Run model and generate predictions
  6. Validation/refinement (useful visuals to aid in this process)

Workflow



Christen, P., Winkler, W.E. (2017). Record Linkage. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7687-1_712



Estimating Model Parameters

…we'll talk about it later

Code

Indiana Department of Health

febrl

- Freely Extensible Biomedical Record Linkage
- Probabilistic RL project (no longer being developed)
- Also published a series of synthetic datasets used to validate RL algorithms
  - febrl4 - “Generated as one data set with 10000 records (5000 originals and 5000 duplicates, with one duplicate per original), the originals have been split from the duplicates, into dataset4a.csv (containing the 5000 original records) and dataset4b.csv (containing the 5000 duplicate records) These two data sets can be used for testing linkage procedures.”

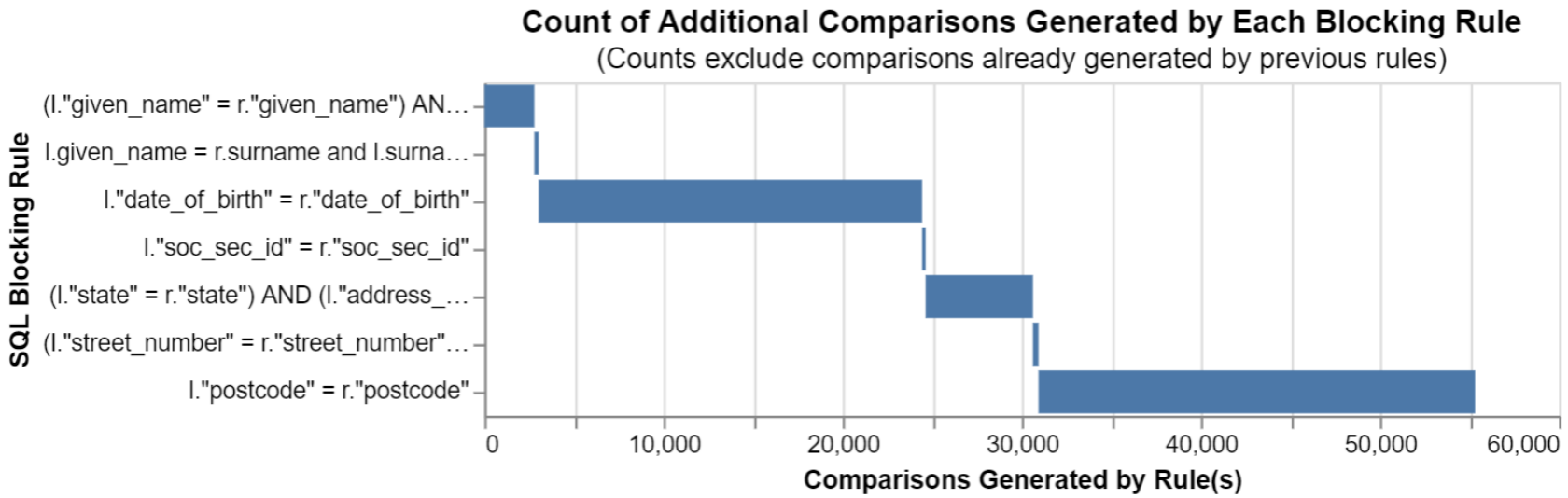
Blocking Rules

"only bother matching candidate pairs if one of these is true"

```
blocking_rules = [
    block_on("given_name", "surname"),
    # A blocking rule can also be an aribtrary SQL expression
    "l.given_name = r.surname and l.surname = r.given_name",
    block_on("date_of_birth"),
    block_on("soc_sec_id"),
    block_on("state", "address_1"),
    block_on("street_number", "address_1"),
    block_on("postcode"),
]
```

5000 (febrl4a) * 5000 (febrl4b) = 25,000,000 comparisons

Σ(comparisons generated by blocking rules)
= 55,314
= 99.8% comparison reduction!



Indiana Department of Health

Settings

- Basic settings
- Blocking rules
- Comparisons
- Supplemental fields

```
# the detailed model considers more columns, using the information we saw in the exploratory phase
# we also include further comparison levels to account for typos and other differences
detailed_model_settings = SettingsCreator(
    unique_id_column_name="rec_id",
    link_type="link_only",
    blocking_rules_to_generate_predictions=blocking_rules,
    comparisons=[
        ctl.NameComparison("given_name").configure(term_frequency_adjustments=True),
        ctl.NameComparison("surname").configure(term_frequency_adjustments=True),
        ctl.DateComparison(
            "date_of_birth",
            input_is_string=True,
            datetime_format="%Y%m%d",
            invalid_dates_as_null=True,
            datetime_metrics=["month", "year", "year"],
            datetime_thresholds=[1, 1, 10],
        ),
        cl.DamerauLevenshteinAtThresholds("soc_sec_id", [1, 2]),
        cl.ExactMatch("street_number").configure(term_frequency_adjustments=True),
        cl.DamerauLevenshteinAtThresholds("postcode", [1, 2]).configure(
            term_frequency_adjustments=True
        ),
        # we don't consider further location columns as they will be strongly correlated with postcode
    ],
    retain_intermediate_calculation_columns=True,
)

linker_detailed = Linker(dfs, detailed_model_settings, database_api=DuckDBAPI())
```

Initialize linker object

Model Parameters

1. λ – the prior probability that any two records match i.e. null case
   - $\lambda = Pr(Records\ match)$
   - Assuming no other knowledge of the data, how likely is a match?
   - Estimate it to be 1/5000, or let splink estimate it (by telling it you think 80% of the matches are true positives)
2. u - probability of a given observation given the records are **not** a match
   - $u = Pr(Observation \mid Records\ do\ not\ match)$
   - Works by sampling random pairs and calculating the distribution of the comparisons, using λ
   - The u probability is a measure of coincidence/cardinality e.g. the chance of having the same last name as a random person is small
3. m - probability of a given observation given the records are a match
   - $m = Pr(Observation \mid Records\ match)$
   - Without training data or a unique ID (e.g. SSN), m must be estimated using an expectation maximization algorithm for all comparisons
   - The m probability is largely a measure of data quality/reliability - if a feature (e.g. DOB) is poorly collected, it may only match exactly for 50% of true matches

```
session_dob = linker_detailed.estimate_parameters_using_expectation_maximisation(
    block_on("date_of_birth"), estimate_without_term_frequencies=True
)
session_pc = linker_detailed.estimate_parameters_using_expectation_maximisation(
    block_on("postcode"), estimate_without_term_frequencies=True
)
```

```
deterministic_rules = [
    block_on("soc_sec_id"),
    block_on("given_name", "surname", "date_of_birth"),
]

linker_detailed.estimate_probability_two_random_records_match(
    deterministic_rules, recall=0.8
)
```

```
INFO:splink.linker:Probability two random records match
is estimated to be  0.000239.
```

```
# We generally recommend setting max pairs higher (e.g. 1e7 or more)
# But this will run faster for the purpose of this demo
linker_detailed.estimate_u_using_random_sampling(max_pairs=1e6)
```

```
WARNING:splink.linker:You are using the default value for `max_pairs`, which
INFO:splink.estimate_u:----- Estimating u probabilities using random sampling

INFO:splink.m_u_records_to_parameters:u probability not trained for date_of_b
INFO:splink.m_u_records_to_parameters:u probability not trained for date_of_b
INFO:splink.m_u_records_to_parameters:u probability not trained for date_of_b
INFO:splink.estimate_u:
Estimated u probabilities using random sampling
INFO:splink.settings:
Your model is not yet fully trained. Missing estimates for:
    - given_name (no m values are trained).
    - surname (no m values are trained).
    - date_of_birth (some u values are not trained, no m values are trained).
    - soc_sec_id (no m values are trained).
    - street_number (no m values are trained).
    - postcode (no m values are trained).
```

https://moj-analytical-services.github.io/splink/topic_guides/theory/fellegi_sunter.html

Validation

Indiana Department of Health

Match Probability

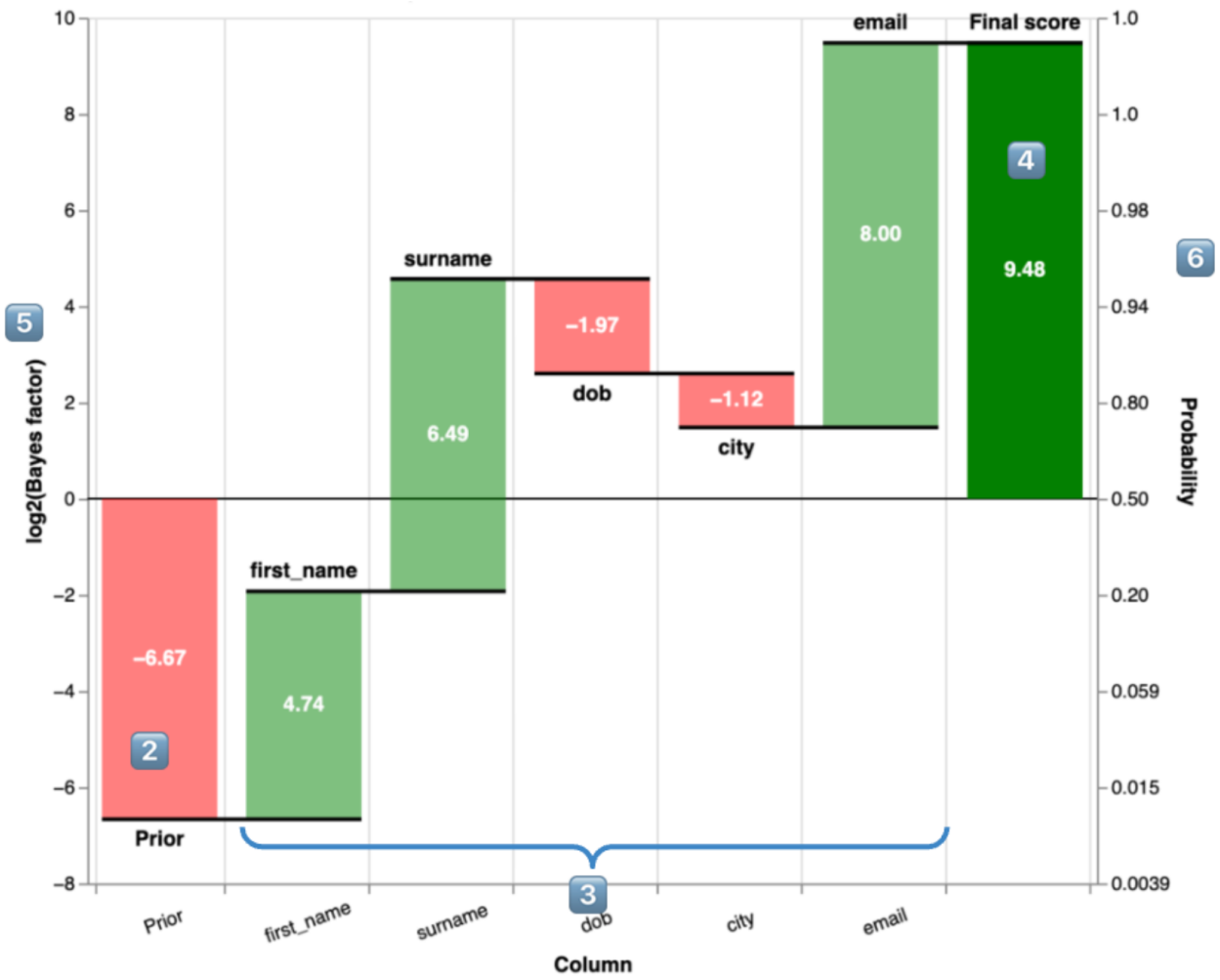
$$match\ weight\ M = log_2\left(\frac{\lambda}{1-\lambda}\right) + log_2\frac{m}{u}$$

$$M_{obs} = M_{prior} + M_{features}$$

1

first_name	▲surname	dob	city	email
Adaam	Jones	1982-10-11	Bristol	ajones6@cortez-wilcox.com
Adam	Jones	1992-09-11	oBristl	ajones6@cortez-wilcox.com

$$\Pr(Match \mid Observation) = \frac{2^{M_{obs}}}{1 + 2^{M_{obs}}}$$



Indiana Department of Health

Generating Predictions

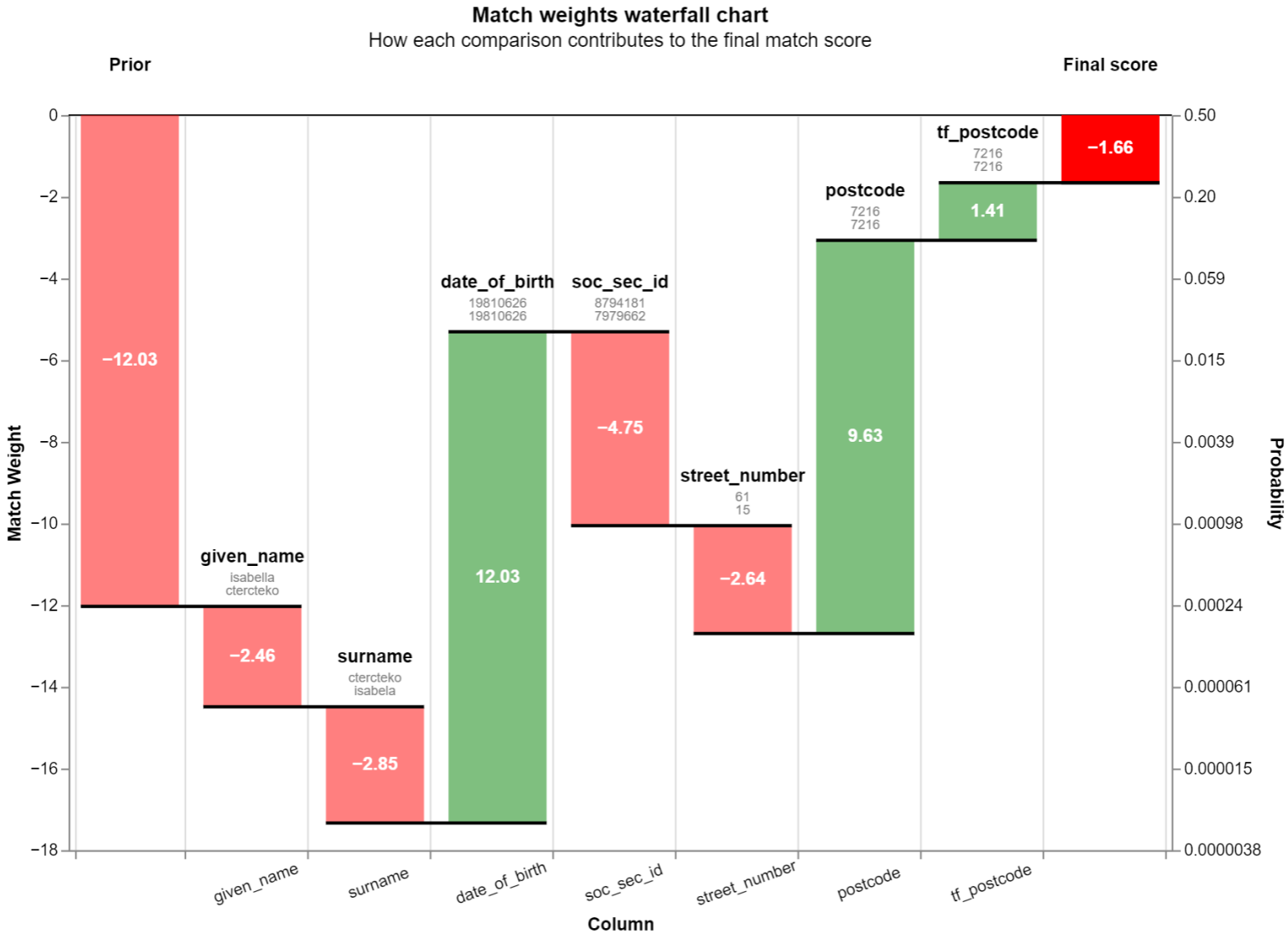
```
predictions = linker_detailed.predict(thresho
df_predictions = predictions.as_pandas_datafra
df_predictions.head(5)
```

```
WARNING:splink.linker:
 -- WARNING --
You have called predict(), but there are some
Comparison: 'date_of_birth':
    m values not fully trained
Comparison: 'date_of_birth':
    u values not fully trained
```

	match_weight	match_probability	source
0	-1.826404	0.219948	__splink__i
1	-1.715576	0.233412	__splink__i
2	-1.170887	0.307550	__splink__i
3	-0.998600	0.333549	__splink__i
4	-0.337082	0.441852	__splink__i

5 rows × 47 columns

```
records_to_view = 3
linker_detailed.waterfall_chart(
    df_true_links.head(records_to_view).to_dict(orient="records")
)
```

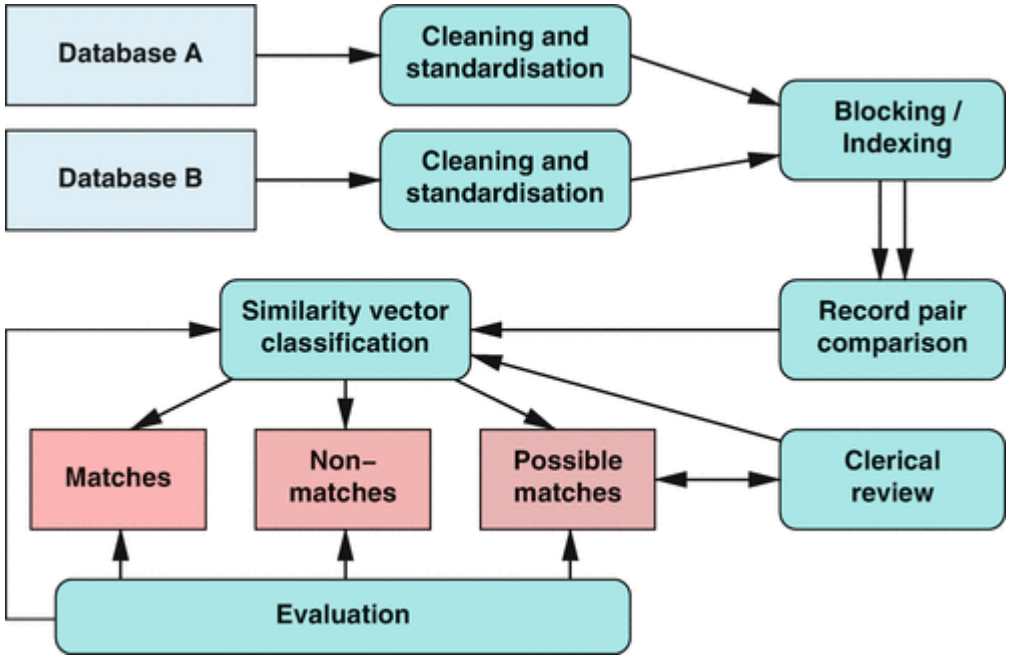


Model Iterations

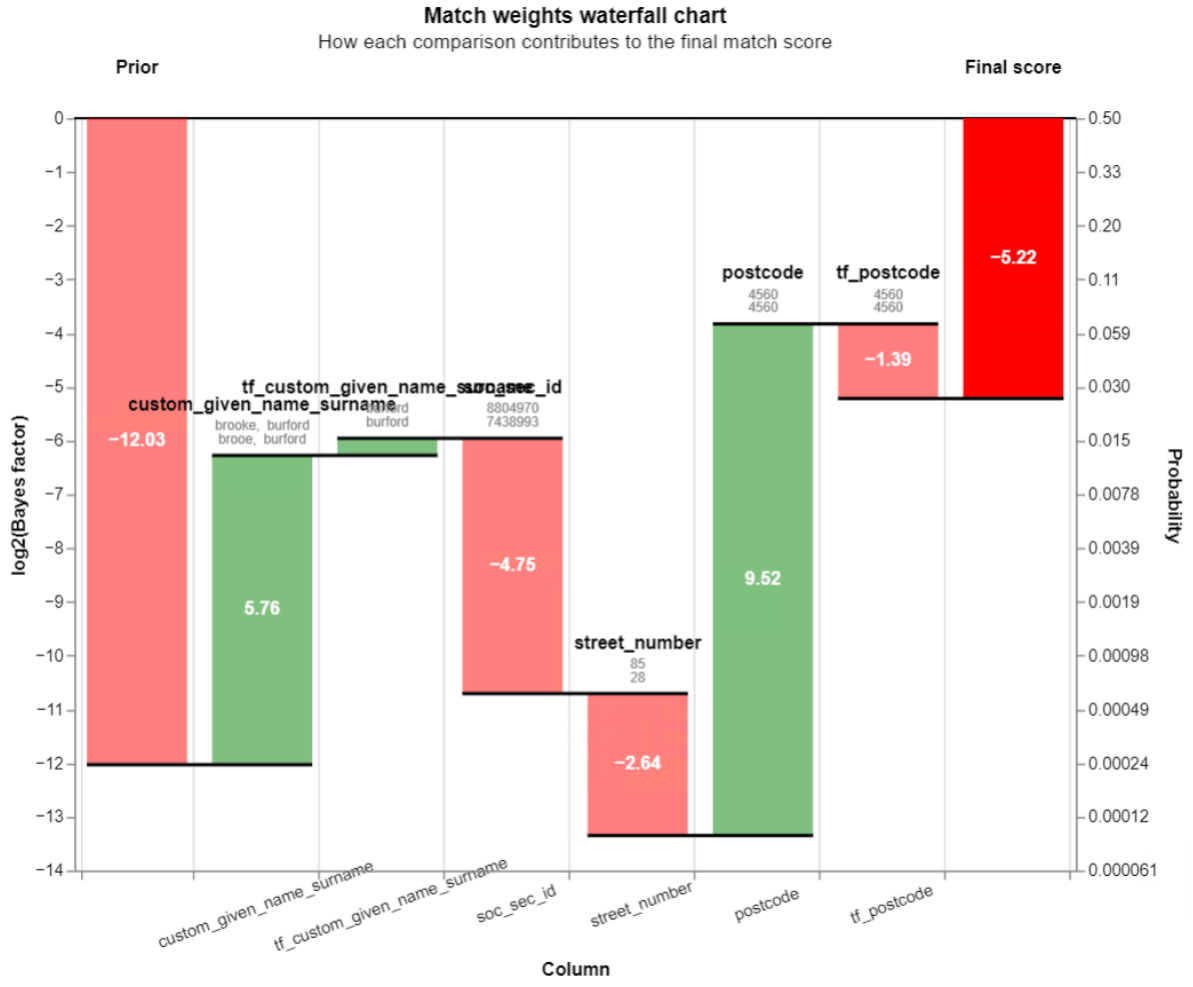
Of the 5000 matching pairs:

- A deterministic match on first-last-DOB = 44%
- A fuzzy match on first-last-DOB = 69.2% (p ~ .9)
- Basic splink settings using first-last-DOB = 72.2% (p = .9)
- Advanced splink settings using first-last-DOB and address = 95.5% (p = .99)
- Optimized splink settings using first-last-DOB, SSN, and address = 99% (p = .99)

Summary



Christen, P., Winkler, W.E. (2017). Record Linkage. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7687-1_712



Indiana Department of Health

Next Steps

Indiana Department of Health

Next Steps

- Additional validation against a large “golden” dataset
- PySpark/Databricks integration (perhaps ARC)
- Create and host boilerplates for wider agency use
- Master Patient Index (MPI)

Acknowledgements

- IDOH Data Science team
- MPH & Resultant
- Dan LaBar and Dan Holder
- Robin Linacre

Additional Resources

- Interactive browser notebook - https://colab.research.google.com/github/moj-analytical-services/splink/blob/splink4_dev/docs/demos/examples/duckdb/febrl4.ipynb
- https://github.com/moj-analytical-services/splink
  `pip install splink` (Python 3.8+)
- Tutorials - https://moj-analytical-services.github.io/splink/demos/tutorials/00_Tutorial_Introduction.html

Questions?

CONTACT:

Matt Simmons

msimmons@health.in.gov