DMI
Data Modernization
INITIATIVE

Illuminating new technologies for capacity and action

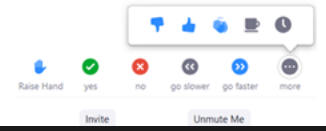
Moderator

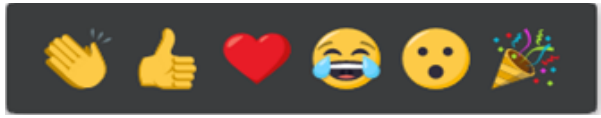
Emilie Lamb, Consultant, Public Health Informatics Institute | May 25, 2022

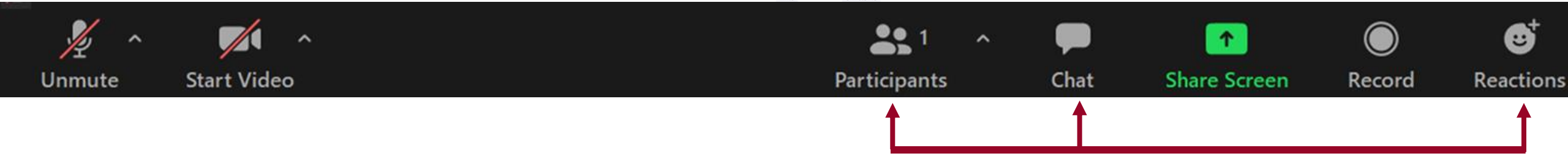
Better data. Better decisions. Better health.

Finding your way around Zoom

- The audience is muted, type your questions into the chat or use reactions to communicate with panelists.







Data Lakes at the Minnesota Department of Health (MDH) - Lessons learned

Aasa Dahlberg Schmit

May 2022

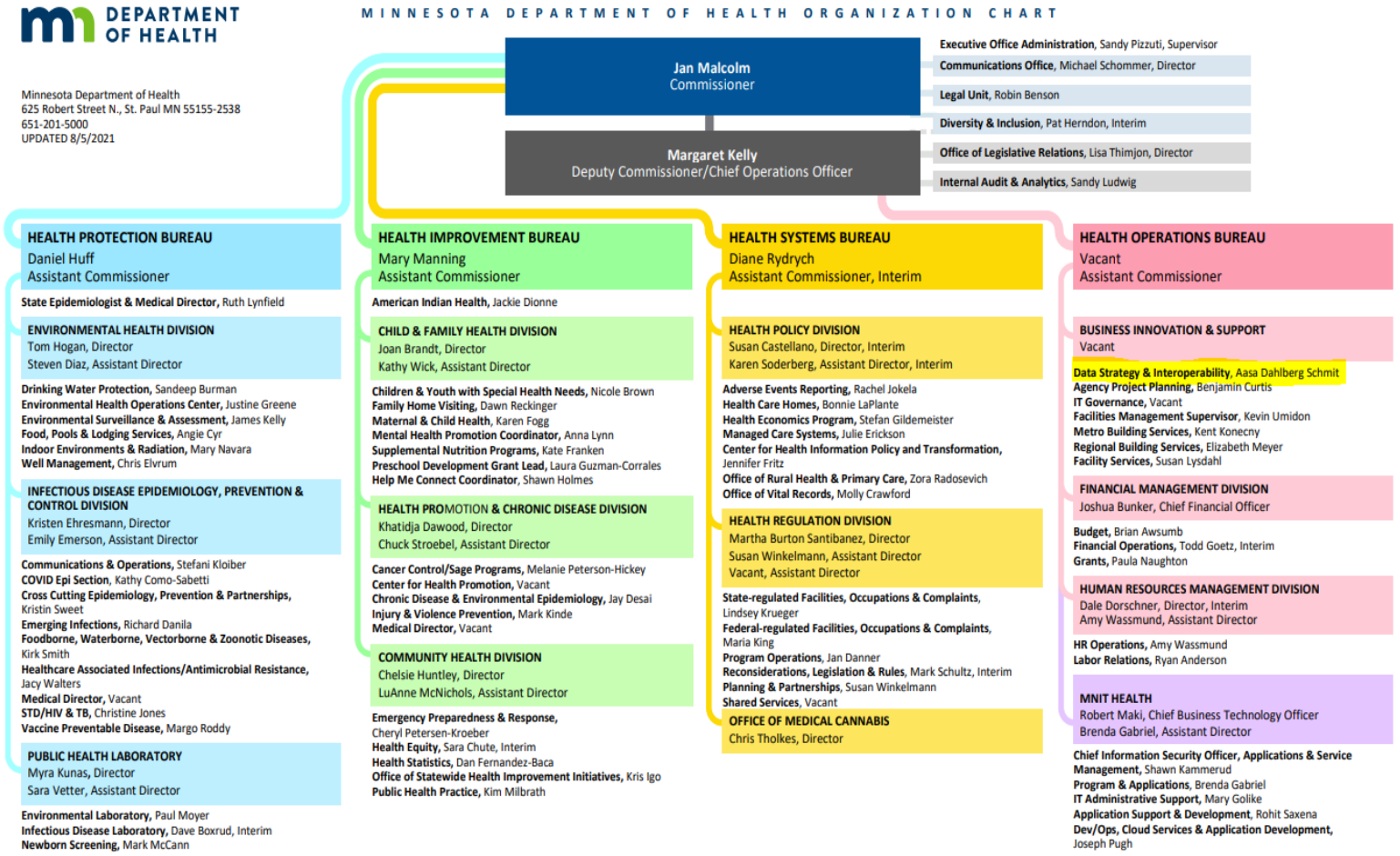
Thanks to

- Sarah Solarz (MDH – MEDSS ops)
- Ann Kayser (MDH – MEDSS ops)
- Stephanie Meyer (MDH – COVID section)
- Joseph Pugh (MNIT)
- Steve Gorg (MNIT)
- Priya Rajamani (MDH – MEDSS ops, UofM)

Overview
Minnesota Department of Health

- Built upon a strong partnership between the Minnesota Department of Health (MDH), local public health agencies, tribal governments and a range of other organizations
- ~ 1,500 employees (not accounting for numerous contractors brought on board for COVID)
- Relies on a mix of state, federal, and other funds
  - for the 2022-2023 state biennial period—from July 1, 2021 to June 30, 2023—the department's projected budget authority is $1.4 billion
- Had informatics activities in varying form since 2008 (pre-meaningful use)
  - With an informatics leader/champion
- Earlier informatics activities were housed in Center for Health Information Policy & Transformation
  - This office is more external facing and coordinates the Minnesota eHealth Initiative

Overview
Office of Data Strategy and Interoperability



- Created in 2019 to lead and drive interoperability efforts at MDH
  - Provides vision, direction, and leadership to advance enterprise data strategies and data exchange with external partners and across MDH programs
  - Oversees federal funding for public health interoperability, develops shared solutions, promotes standards and ensures security and privacy in data systems
  - Direct reporting of DSI Director to the Deputy Commissioner of Health and connections with Executive Leadership

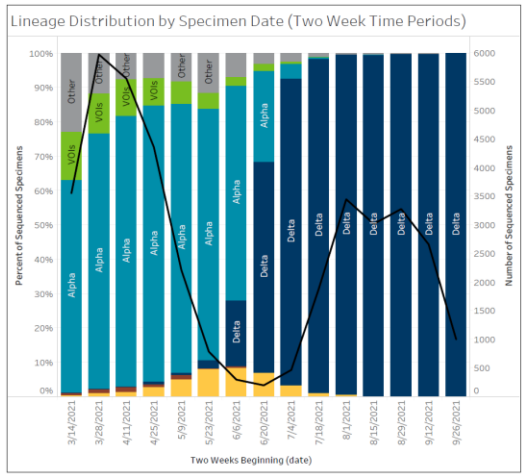
Data Management at MDH at start of COVID

- Typical investigation can use smaller tools
  - Access
  - REDCap
  - Excel

All allow a user to design, change, manage without substantial information technology infrastructure needs

- Larger investigations and standard disease surveillance require larger tools with IT infrastructure
  - MEDSS (Minnesota Electronic Disease Surveillance System)
  - MIIC (Minnesota Immunization Information Connection)
  - Minnesota Vital Records (death certificates, birth certificates)
- All IT systems was moved to the Amazon cloud in 2018-2019

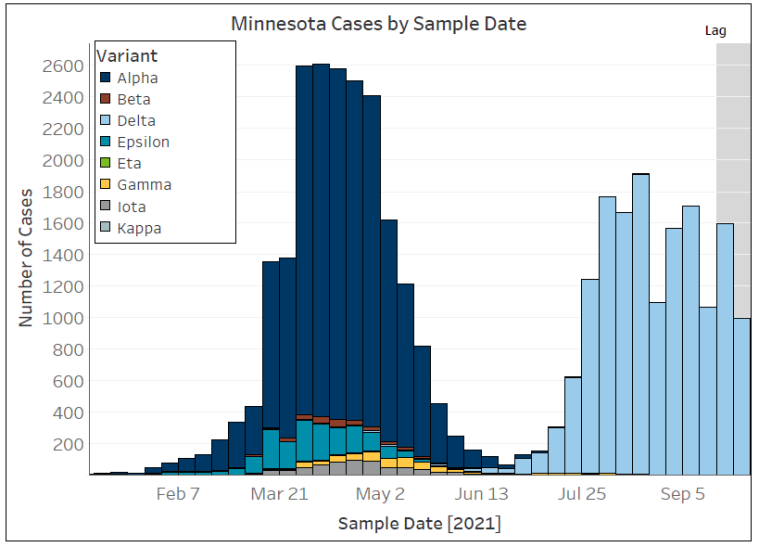
Pre-Data Lake Challenges
Continual Need for Data Reporting



Variant Distribution Since 8/22/2021*		
Delta	B.1.617.2	84.99%
	AY.3	14.79%
	AY.1	0.04%
Gamma	P.1	0.01%
Other	Other	0.17%

Variants of Concern (VOCs)	
B.1.1.7 (Alpha)	18,227
B.1.351 (Beta)	262
P.1 (Gamma)	552
B.1.617.2 (Delta)	15,858

Variants of Interest (VOIs)	
B.1.427 & B.1.429 (Epsilon)	1,778
B.1.525 (Eta)	35
B.1.526 (Iota)	609
B.1.617.1 (Kappa)	4



Race/Ethnicity	
White	30,619
Black	1,521
Asian	1,402
American Indian/Alaska Native	442
Native Hawaiian/Other Pacifi..	42
Multiple Races	514
Hispanic	1,425
Other	536
Missing/Unknown	2,307

Interview Status	
Interviewed	11,335
LTF	22,980
Refused	836
Pending	3,657

Re-Interview Status	
Interviewed	2,599
LTF	1,224
Refused	341
Pending	34,640

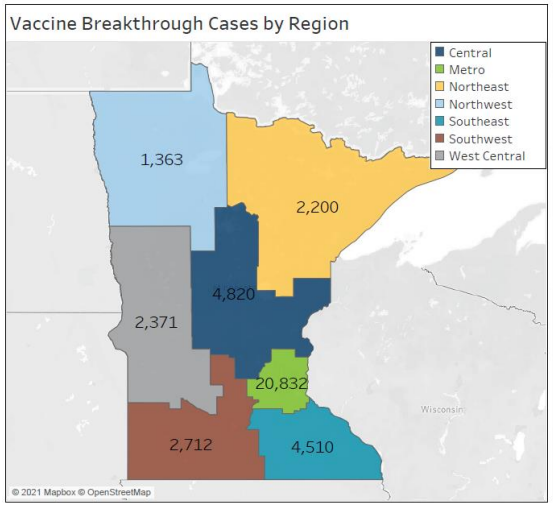
Gender	
Female	21,795
Male	16,701
Other	11
Unknown	301

Median Age (Min, Max):
49.0 (11, 106)

Variants	
B.1.1.7 (Alpha)	779
B.1.617.2 (Delta)	4,286
B.1.351 (Beta)	18
P.1 (Gamma)	53
VOI	110
Other	115
None	748
Not Sequenced	32,699

Age Groups	
10-14	342
15-19	1,197
20-24	2,016
25-29	2,765
30-34	3,179
35-39	3,570
40-44	3,707
45-49	2,914
50-54	3,056
55-59	3,079
60-64	3,158
65-69	3,011
70-74	2,333
75-79	1,794
80-84	1,354
85-89	803
90-94	378
95-99	134
100+	18

Underlying Health Conditions	
Any underlying health condition	5,240
Former smoker	1,950
Obesity	1,446
Heart Conditions	1,065
Cancer	770
Renal/kidney disease	351
Current smoker	499
Immunosuppressive medications	324
Emphysema/COPD	285
Solid organ transplant	83
Severe obesity	236
Sickle cell	28
Down Syndrome	14
Other underlying medical condition	2,716



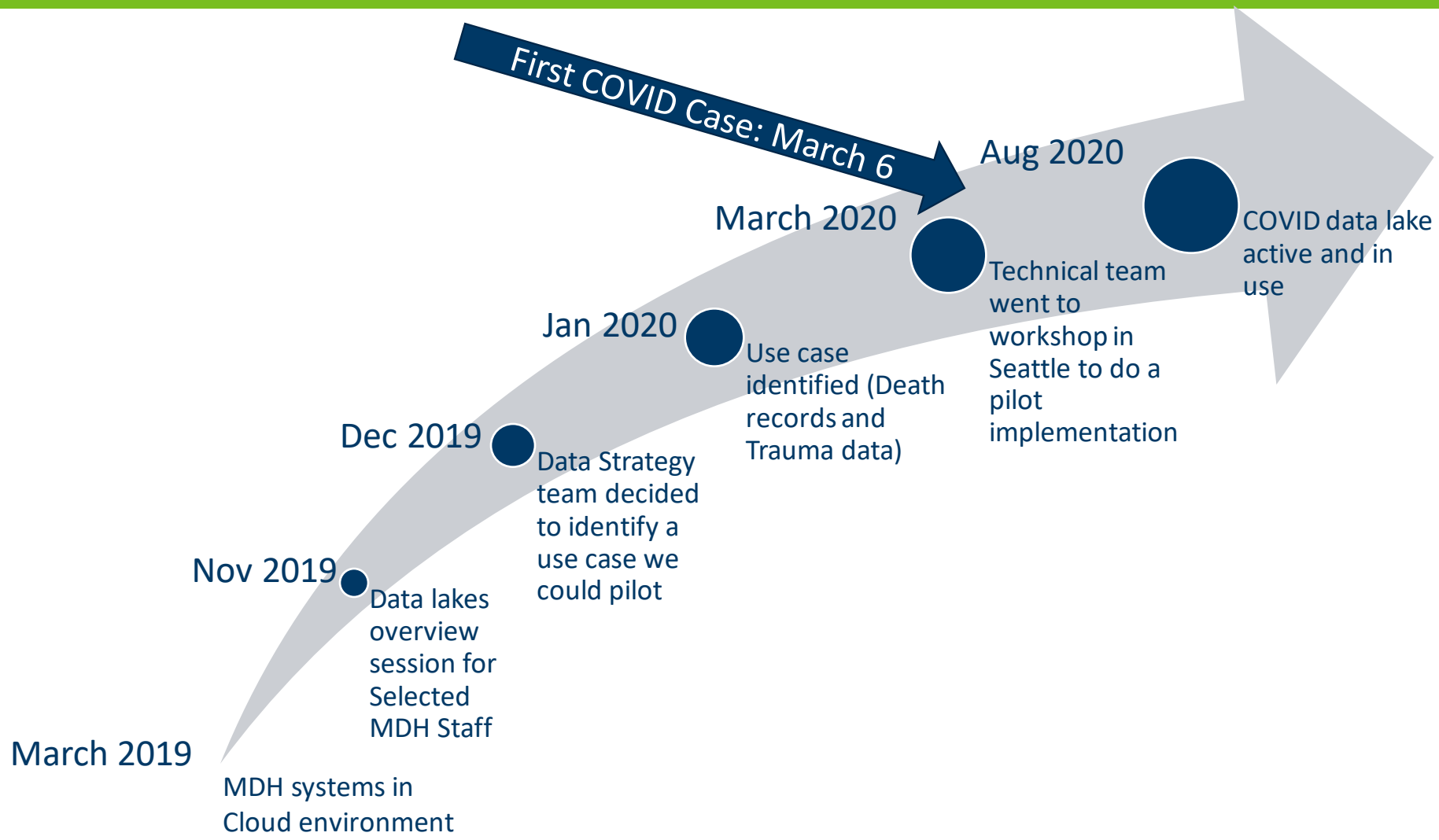
Central	4,820
Metro	20,832
Northeast	2,200
Northwest	1,363
Southeast	4,510
Southwest	2,712
West Central	2,371

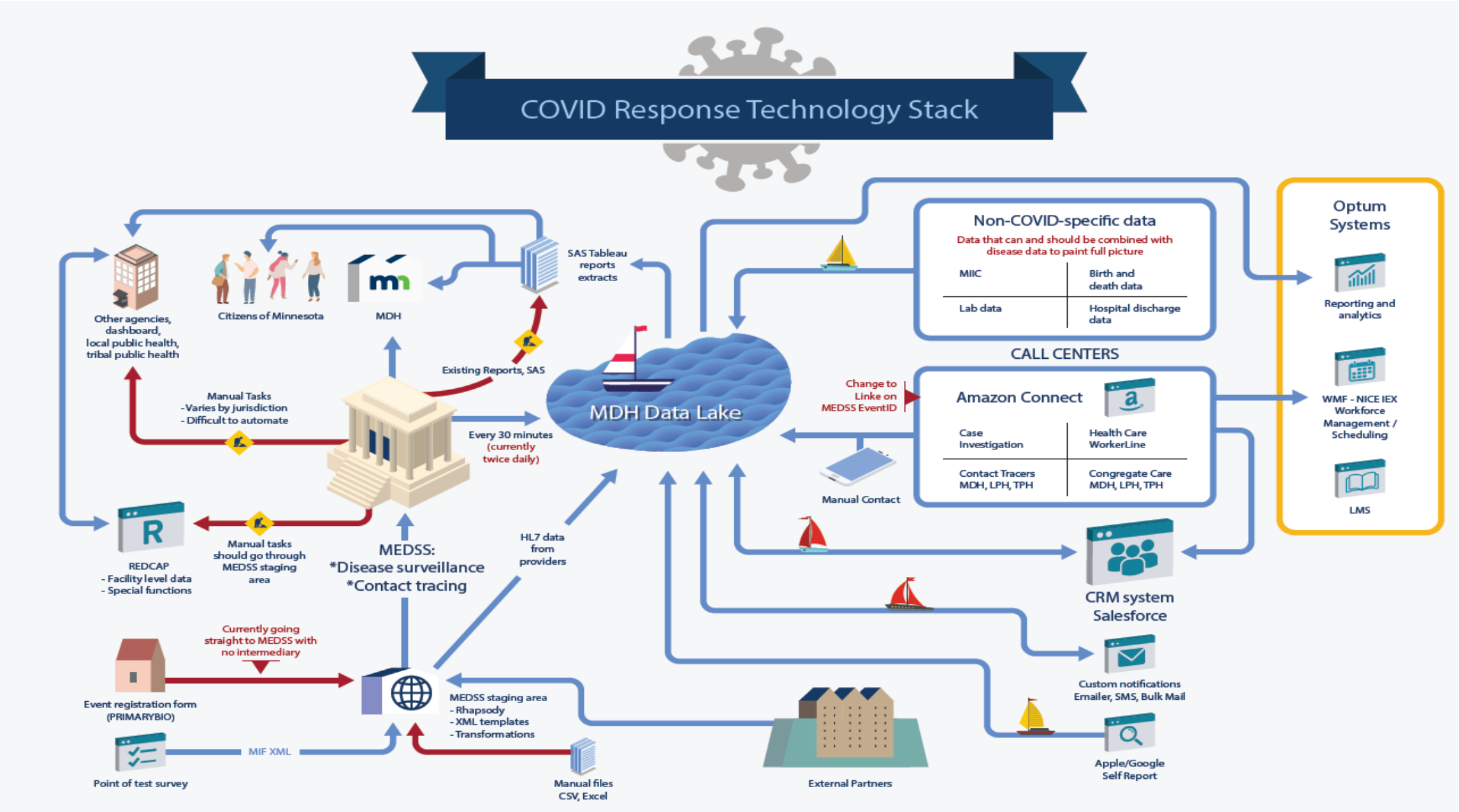
DEPARTMENT OF HEALTH

Pre-Data Lake Challenges
Keeping up with New Tools

- Changes to allow for additional data feeds or additional data elements required investments, especially IT staff/admin time
- Siloed data with different purposes and uses, combination of datasets was happening outside of the systems (for example by downloading sets locally and merge them) – very time consuming
- To expose our data to new tools and interfaces would require changes to source systems

Move to Data Lakes for COVID Response at MDH



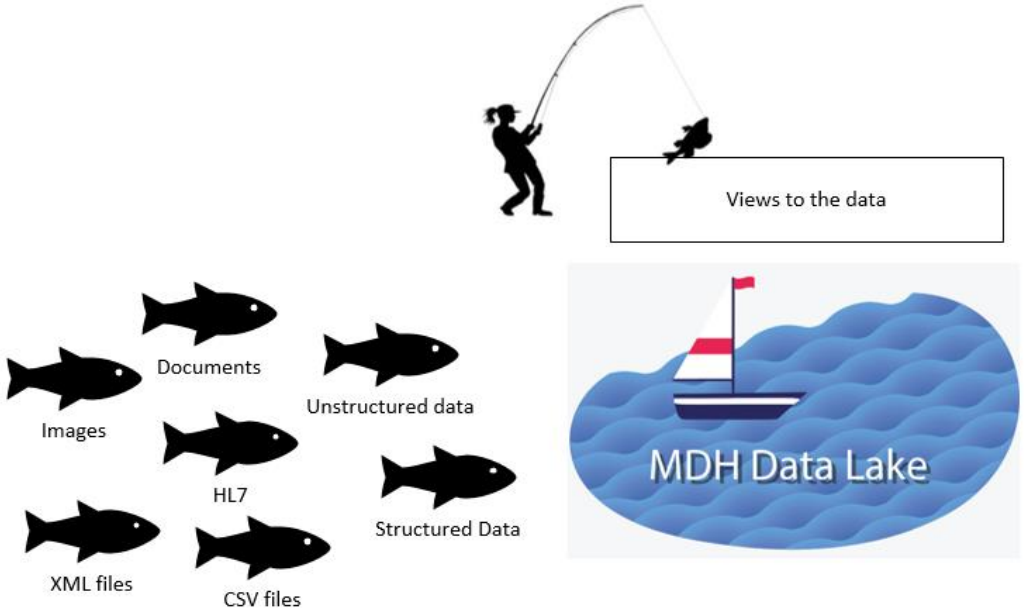

COVID Response Technology Stack
Other agencies, dashboard, local public health, tribal public health
Citizens of Minnesota
MDH
SAS Tableau reports extracts
Existing Reports, SAS
Manual Tasks
-Varies by jurisdiction
- Difficult to automate
Every 30 minutes (currently twice daily)
MDH Data Lake
Non-COVID-specific data
Data that can and should be combined with disease data to paint full picture
MIIC
Birth and death data
Lab data
Hospital discharge data
CALL CENTERS
Change to Linke on MEDSS EventID
Amazon Connect
Case Investigation
Health Care WorkerLine
Contact Tracers MDH, LPH, TPH
Congregate Care MDH, LPH, TPH
Manual Contact
Optum Systems
Reporting and analytics
WMF - NICE IEX Workforce Management / Scheduling
LMS
REDCAP
- Facility level data
- Special functions
Manual tasks should go through MEDSS staging area
MEDSS:
*Disease surveillance
*Contact tracing
HL7 data from providers
CRM system Salesforce
Custom notifications Emailer, SMS, Bulk Mail
Currently going straight to MEDSS with no intermediary
Event registration form (PRIMARYBIO)
MEDSS staging area
- Rhapsody
- XML templates
- Transformations
MIF XML
Point of test survey
Manual files CSV, Excel
External Partners
Apple/Google Self Report

MDH Data Lakes: Benefits

- Reporting platform from a single source
- Fast ingestion and consumption of disparate data sources
- Increased speed and efficiency
- Ability to automate allowed for rapid experimentation and scalability
- Leveraging serverless technologies has led to reliability
- Improved collaboration
- Ability to access data outside of our legacy systems
  - Lot of our data quality issues was caught when looking at the data in the data lake
  - When we had a backlog with intake, we could look at numbers before they completed those steps

MDH Data Lakes: Challenges

- It was easy to add data, not as easy to access and use
  - Had to learn new ways, terminology and tools
- New skills needed, both IT to learn the technology and MDH to learn how to access the data and how to use it
- Initially focus on feeding data to other applications (such as CICT and texting), use and query of data didn't get as much attention
- Uncertainty of what's happening with the data and who can access it (lack of governance)
- Legacy system and data lake had different standards and data dictionaries (for example date fields)
- Competing priorities



MDH Data Lakes: Lessons learned

- DSI office allowed for quick decision making
- Difference between a normal data repository (where you build the schema first) and the lake (where you have to add the schema/views) after – Needs special skillset
  - Skills shift from IT to MDH, new ways of interacting with the data
  - Other people in IT than what we were used to working with
- More conversations and documentation up-front when data was added to the lake and when views was created would have helped (creating naming standards etc)
- Lake need Governance and ownership
- AWS participation and assistance was critical
- Once we started, additional needs was identified quickly, enforcing the need for governance

MDH Data Lakes: Lessons learned

If we were to select the top 3 most important items to make the MDH Data lakes a success for the future, what would it be?

- Knowledge sharing and training of MDH staff
- Better documentation and data dictionaries
- Governance

Thank you

Aasa Dahlberg Schmit: aasa.dahlberg.schmit@state.mn.us

An Approach to Batch Master Patient Id Assignment

Mark Alexander

New York City Department of Health and Mental Hygiene

May 25, 2022

Purpose of the Patient Matching Project

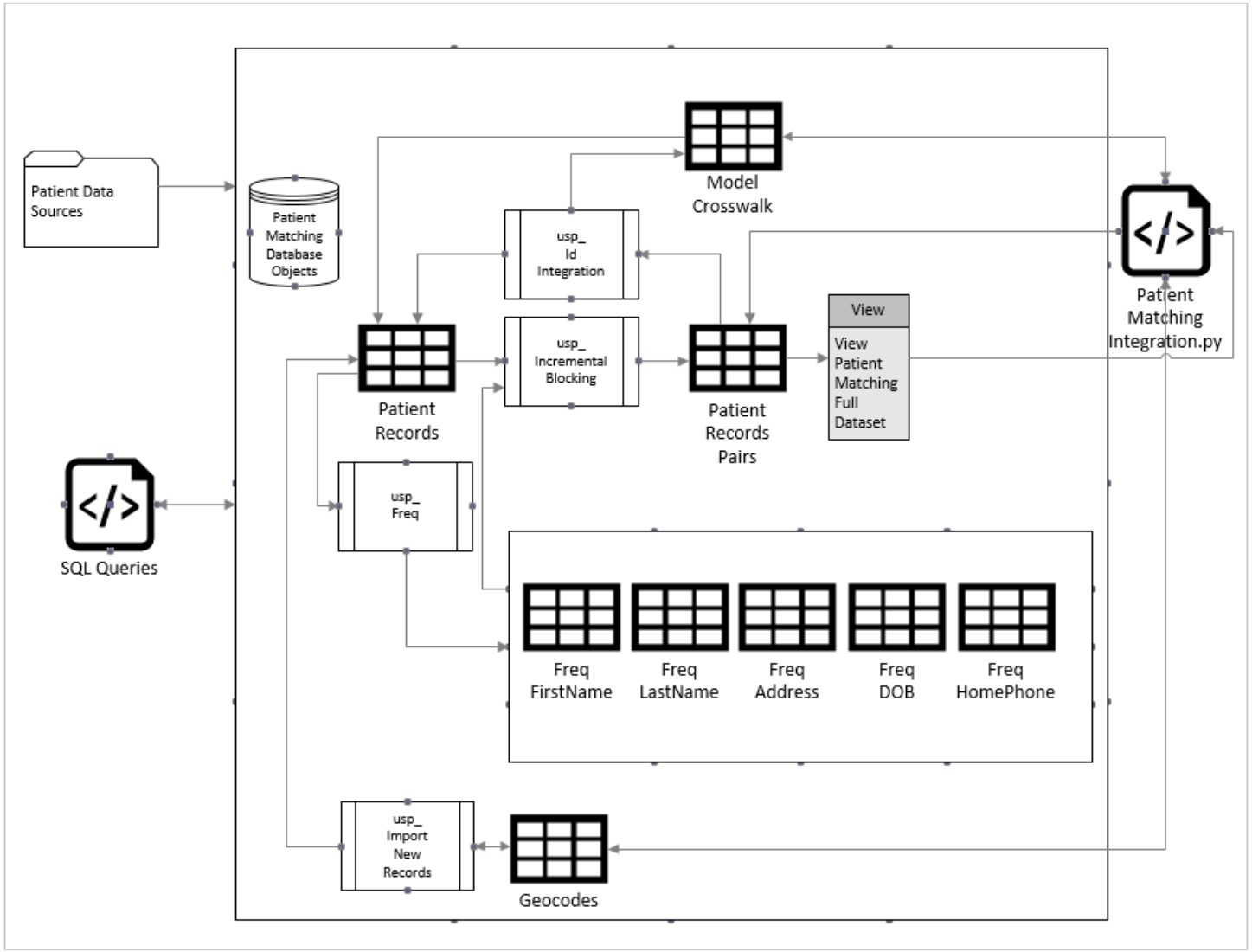
- ***To explore, evaluate, and apply technologies for matching patient data across NYC DOHMH to offer agency wide solutions***

- If a technology is adopted, interfaces can be customized for analysts to leverage
  - Webservices that can be applied in a script
  - Batch matching interface that can return data linked to internally assigned patient id

Mechanics of Implementation

Internally Developed Process

- Using Electronic Case Reporting (ECR) and Electronic Clinical Lab Reporting System (ECLRS) data as test systems, a batch matching process was built
- Using the agency's Linux Python server and a SQL Server database, a process to match newly received patient records against ~3 years of patient records (~60 Million records) from those systems was built for a limited scope Master Patient Index
- Python script is put on cron job on server
  - Script interacts with SQL Server database via stored procedure
  - SQLAlchemy python library used to efficiently write python pandas records to SQL Server database tables



Basics of Approach

2.
Pairs of records are created basic on join logic

3.
Fields are constructing to compare records in pairs

Patient Records

Id	First Name	Last Name	DOB	Sex	MPID
1	Mark	Alexander	1/22/1990	M	1
2	Mark L	Alexander	1/1/1990	M	1

Blocking

Id Reconcile

Paired Records & Modeling Features

Id1	Id2	First Name Comparison	Last Name Comparison	DOB Exact	DOB Comparison	Sex Exact	Match
1	2	0.89	1.0	0.0	0.87	1.0	1

1.
Records are added to table of demographic, contact, and id data

Validation is carried out to clean native fields and make derivative ones necessary for matching

5.
Pairs of ids are reconciled to determine which ids roll up to the same person and lowest id selected as Master Patient Id to link to original source ids

4.
Model applies score to pairs and determines whether records are matches or not

Importing Records

Considerations

- Incorrectly populated fields
- Default values or values populated with facility data
- Non-standardized addresses
- Message based systems tend to create duplicate records

Approaches

- Validation functions to populate fields
- Analysis of frequencies to find likely default or facility values
- Geocoding of records where possible
- A deterministic “naïve” deduplication step to represent many records with one

Concatenating fields and hashing string, a naïve deduplicated value can be referenced with the lowest PatientRecordId assigned to matching rows and boolean denoting which record is original populated

PatientRecordId	System	NativeSystemId	FirstName	LastName	DOB	Address	Phone	Email	GeocodedAddress	NaiveDedupePatientRecordId	NaiveDedupe
1	1	14223164	Mark	Alexander	1/22/1990	58 Farthington rd	5165554321	mark.alexander@vmail.com	58 Farthington Road	1	1
2	1	17946349	Mark	Alexander	1/22/1990	58 Farthington road	5165554321	mark.alexander@vmail.com	58 Farthington Road	1	0
3	2	13891674	Mark	Alexander	1/22/1990	58 Farthington rd	5165554321	mark.alexander@vmail.com	58 Farthington Road	1	0
4	1	17001075	Mark	Alexander	1/1/1990	58 Farlington rd	5165554321		58 Farthington Road	4	1
5	2	11982166	Mark	Alexander	1/1/1990	58 Farthington rd	5165554321		58 Farthington Road	4	0
6	1	19403406	Mark	Alexander	1/1/1990	58 Farthington rd	5165554321		58 Farthington Road	4	0

All messages from ECR and ECLRS are assigned an autoincrementing integer Id while native system ids are captured in table

Geocoding is written back to table and building number and street info parsed to separate fields

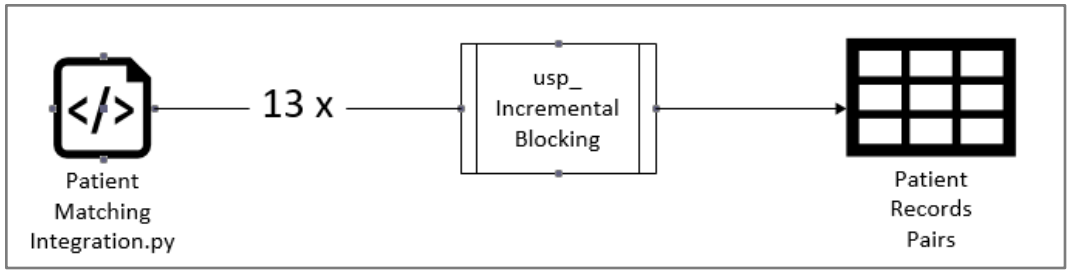
Blocking

- **Blocking is the process of creating pairs of records for a matching algorithm to evaluate**
- Records are paired together based on rules
  - Rules too restrictive leave out pairs that should be matched, but have dissimilarities
  - Rules too loose will add too many irrelevant pairs
- Incremental blocking only compares newly imported records with themselves and previously matched records
- Only NaïveDedupe = 1 records are used in blocking step

Blocking Rules (Union of All)

1. Phonetic Name, DOB Components
2. Contained Name, DOB Components
3. Name Tokens, DOB Components, Sex
4. Phonetic or Contains Name Switch, DOB Components
5. SSN
6. Email, DOB Components
7. Home Phone, DOB Components
8. Work Phone, DOB Components
9. EMR IDs
10. Building No, Zip, DOB Components
11. Street, Zip, DOB Components
12. NGrams, DOB Components
13. Partial SSN Last 4 Digits, DOB Components

Python script multithreads calls to blocking stored procedure running all blocking rules in parallel



Comparisons

Central question:

- What helps determine whether two records belong to the same person?

Approach

- Prepare a variety of features that measure dissimilarity of fields
- Include indicators of data quality and meta data to lend context to dissimilarities
- Blocking stored procedure makes simple comparison features; complex string comparisons are carried out by python server

Naïve Score

Base Score

0.4*(Name Composite) +
0.2*(DOB Match) +
0.1*(Sex Match) +

Confirmatory Score

0.3*(SSN Match) or
0.3*(Address Match) or
0.3*(Email Match) or
0.3*(Home Phone Match)

Naïve score enables high level comparison of pairs prior to machine learning model

Sample of Features

Category	Features	Calculation
Demographics	First Name	Jaro Winkler, first letter match, token match
	Last Name	Jaro Winkler, first letter match, token match
	Sex	Exact match
	DOB	Exact match, Damerau Levenshtein, Date Difference
	Zip Code	Exact match, Damerau Levenshtein
	Phone	Exact match, Damerau Levenshtein
	Building No	Exact match, Damerau Levenshtein
	Street	Jaro Winkler
	SSN	Exact match, Damerau Levenshtein
	Email	Exact match
	Race	Exact match
	Ethnicity	Exact match
	Hispanic	Min across pair, Max across pair
	Asian	Min across pair, Max across pair
Data Quality	Invalid DOB	Min across pair
	Invalid SSN	Min across pair
	Invalid Email	Min across pair
	Invalid Phone	Min across pair
	One Letter Name	Min across pair
	Name Switch	Exact match
Meta Data	Record Date	Date difference
	First Name Freq	Max across pair
	Last Name Freq	Max across pair
	DOB Freq	Max across pair

Applying Modeling

- Selected model is applied to all candidate pairs to classify them as matches or non matches
  - Model is stored as pickle object on server for reference
- Matched pairs can be rolled up to person level entities and written back to PatientRecords table

Matched Id Pairs

Patient RecordId 1	Patient RecordId 2	Match
1	2	True
2	5	True
2	3	True
3	4	True
4	5	True

All Patient Records Rolled up to Minimum Related Id (Master Patient Id)

Master Patient Id	Patient RecordId
1	1
1	2
1	3
1	4
1	5

Python process to recursively join matched id table to itself unwinds collection of pairs to lowest level patient record id

Modeling

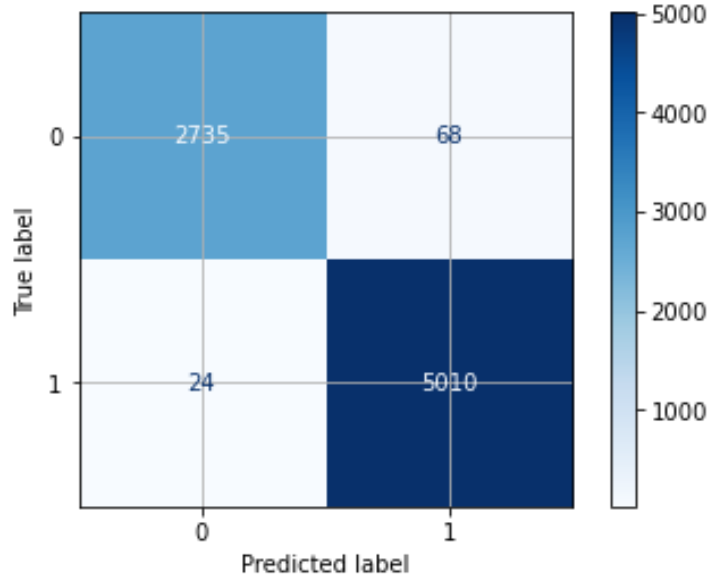
Building Training Dataset

- Small amount of records verified as matches by agency surveillance tool contributed to training dataset
- Additional training data was manually labelled by:
    1. Discretizing main comparison fields and taking random sample of records proportionate to share of panel of overall blocked pairs
    2. Discretizing pairs by naïve score and taking random selection of pairs within score range
    3. Selecting extreme scores of pairs based on query logic, manually inspecting panels, and classifying all pairs within panel

Modeling Process Summary

- Training dataset is divided into training and testing sets with 80-20 split ratio
- GridSearchCv is used to find optimal values of hyperparameters for a range of classifiers on training data
- Models are fit on training data and their classification performance is compared on test data
- Precision, Recall and F1 score are used as evaluation metrics
  - Precision: What proportion of predicted positives is truly positive? $\frac{TP}{TP+FP}$
  - Recall: What proportion of actual positives is correctly classified? $\frac{TP}{TP+FN}$
  - F1 score: Harmonic mean of precision and recall $2 * \frac{Precision * Recall}{Precision+Recall}$

Classifier	Precision	Recall	F1 Score
Extra Trees Classifier	0.986609	0.99523	0.9909
Random Forest Classifier	0.98583	0.99503	0.99041
Gradient Boosting Classifier	0.98403	0.99146	0.98773
Ada Boost Classifier	0.98435	0.98709	0.98572
Logistic Regression	0.981299	0.99027	0.98576
Decision Tree Classifier	0.987156	0.97716	0.98213
K Nearest Neighbors Classifier	0.979859	0.99543	0.98758
Ridge Classifier	0.974445	0.9847	0.97955
Naive Bayes	0.642338	1	0.78222
Quadratic Discriminant Analysis	0.642338	1	0.78222



Current Model Results

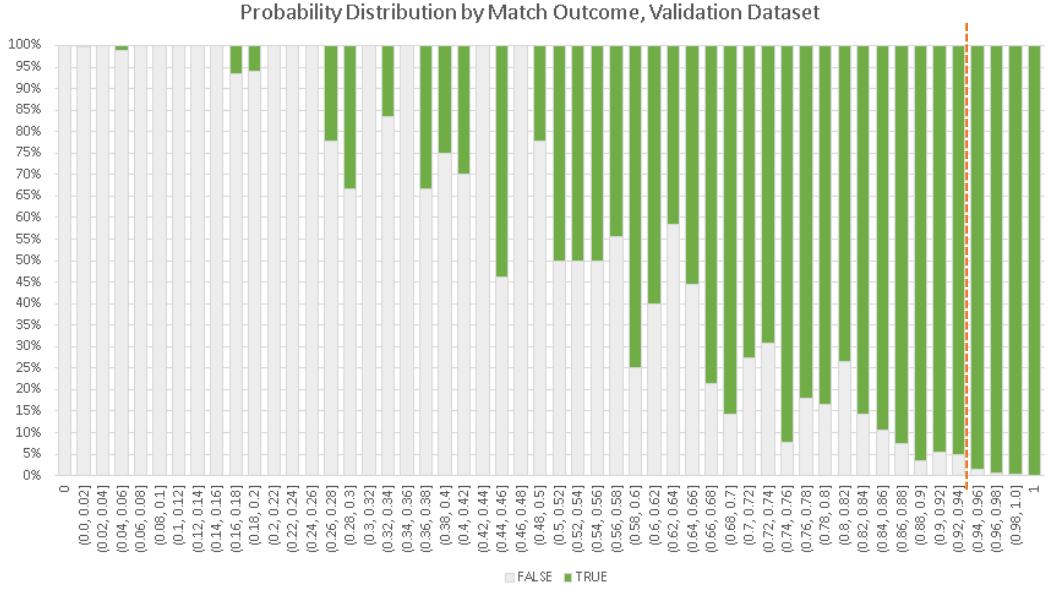
ECR & ECLRS 2020

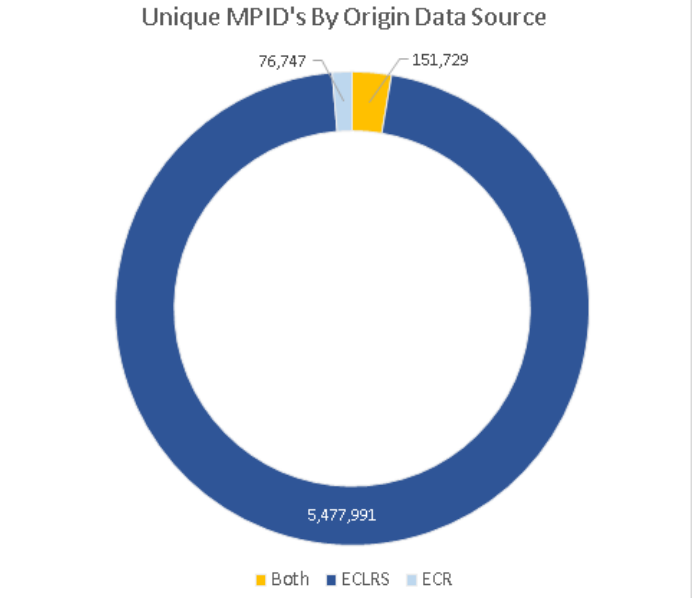
Validation Dataset	Precision	Recall	F1 Score	False Positive Rate	False Negative Rate
Random Forest Classifier	0.99986	0.993557	0.996701	0.000429	0.006443
Extra Trees Classifier	0.99984	0.993815	0.996820	0.000495	0.006185
Gradient Boosting Classifier	0.99941	0.995201	0.997305	0.001850	0.004799

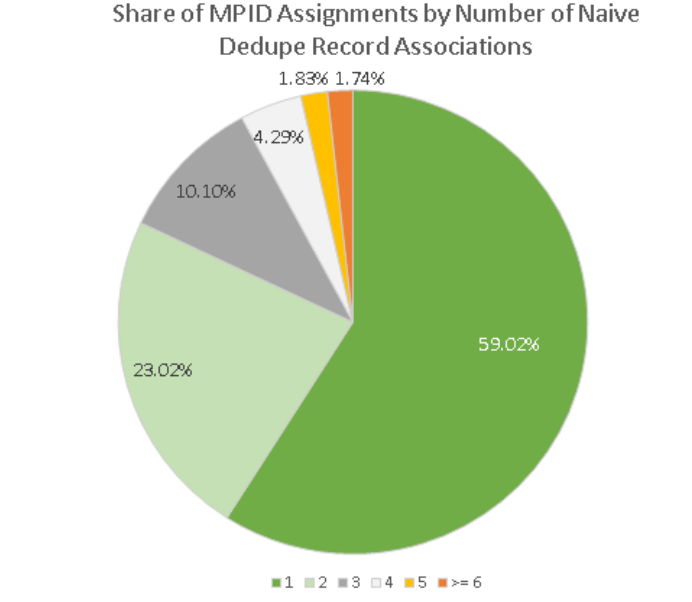
Defining a probability threshold for classification to minimize false positives will greatly reduce merging of record sets across patients

	ECLRS	ECR	Total
Number of Records	17,960,955	499,394	18,460,349
Number of Unique Patients Identified	5,629,720	228,476	5,706,467

Score	Feature	Score	Feature
0.146966	Name_Comparison	0.001168	LastName_FreqMax
0.118632	LastName_Comparison	0.000871	SSN_Edits
0.106960	Sex_Exact	0.000464	SSN_Exact
0.092282	FirstName_Comparison	0.000441	SSN_EitherInvalid
0.064917	ZipCode_Exact	0.000411	SSN_EitherNotPopulated
0.058109	DOB_Exact	0.000382	ZipCode_EitherNotPopulated
0.043434	Sex_EitherNotPopulated	0.000380	Street_EitherNotPopulated
0.043268	DOB_AbsDiffDays	0.000359	Minors_Both
0.035570	DOB_Edits	0.000243	DataSource_Exact
0.033142	Address_FreqMax	0.000227	BuildingNumber_EitherNotPopulated
0.032591	HomePhone_Edits	0.000199	Datasource_ECR
0.030383	BuildingNumber_Edits	0.000181	Ethnicity_EitherNotPopulated
0.024556	NameSwitch_Comparison	0.000148	HomePhone_EitherInvalid
0.022845	LastNameFirstName_Comparison	0.000132	FirstName_OneLetter
0.020091	DOB_Comparison	0.000114	Ethnicity_Exact
0.018477	BuildingNumber_Exact	0.000101	Race_Exact
0.016550	DOB_FreqMax	0.000081	Email_Exact
0.016191	HomePhone_FreqMax	0.000065	Hispanic_Either
0.015608	HomePhone_Exact	0.000063	Race_EitherNotPopulated
0.011944	FirstNameLastName_Comparison	0.000051	MiddleInitial_Exact
0.011916	Street_Comparison	0.000049	Email_EitherNotPopulated
0.005084	Street_Exact	0.000046	FirstInitialMiddleInitial_Exact
0.004970	RecordDate_AbsDiffDays	0.000040	Asian_Either
0.003788	StreetNumber_Exact	0.000033	MiddleName_EitherNotPopulated
0.003585	Female_Both	0.000024	LastName_OneLetter
0.003427	HomePhone_EitherNotPopulated	0.000017	Hispanic_Both
0.002778	StreetNumber_EitherNotPopulated	0.000017	Datasource_ECLRS
0.002663	Minors_Either	0.000009	Asian_Both
0.001726	Female_Either	0.000003	Email_EitherInvalid
0.001225	FirstName_FreqMax		







Looking Forward

Building Resources to Share Across Agencies, pt 1

1. Completing and sharing standalone python library that can carry out bulk deduplication specifically for patient matching
   - Without having to deploy database resources, library functions can accept a dataframe, a feature weighting dictionary, and score threshold, and the records within that dataframe can be assigned an MPID
   - Current record linkage solutions can't function on large datasets (this solution is being tested on 40 millions records) and aren't designed for specifically to match patients

2. Completing and sharing GUI for selecting record pairs from dataframe for labelling to build training dataset
   - This draws on the library developed to carry out blocking and calculates large range of comparisons once user has manually inspected records and assigned labels

Building Resources to Share Across Agencies, pt 2

3. Building shared repo of anonymized pair level dissimilarity/similarity training dataset that analysts across agencies can model with
   - Doesn't include actual patient data, but exhaustive dissimilarity/similarity features based on pairs of real records
   - Pools collective effort of labelling work to maximize benefit and minimize labor
   - Prior project code discussed will be incorporated into solution
   - Code to create dissimilarity/similarity features will be released publicly

Building Resources to Share Across Agencies, pt 3

- Below are a sample of the sort of features that could released with a match status
- The function call to the right is how you could generate the output below

```
master_comparison(
    firstname_1 = 'Lovedeep',
    firstname_2 = 'Marcos',
    middlename_1 = 'Alvarez',
    middlename_2 = 'Alexander',
    lastname_1 = 'Bajaj',
    lastname_2 = 'Alvarez',
    sex_1 = 'Male',
    sex_2 = 'M',
    dob_1 = datetime(1987,1,22),
    dob_2 = datetime(1987,1,1),
    address_1 = '3045 Hobart St',
    address_2 = '30-45 Hobart Street',
    zip_1 = '11372',
    zip_2 = '10520',
    phone_1 = '5166706410',
    phone_2 = '5166784492',
    ssn_1 = '062458596',
    ssn_2 = '062458596',
    email_1 = 'mark.lawrence.alexander@gmail.com',
    email_2 = 'mark.lawrence.alexander@yahoo.com',
    mrn_1 = '1235189',
    mrn_2 = '123518968168'
)
```

'MatchStatus': 'Match',	'lastnamefirstname_eithernull': False,	'middlenamefirstname_eithernull': False,	'dob_eithernull': False,	'ssn_eithernull': False,
'firstname_eithernull': False,	'lastnamefirstname_comparison_exact': False,	'middlenamefirstname_comparison_exact': False,	'dob_comparison_levdam': 2,	'ssn_eitherinvalid': False,
'firstname_comparison_exact': False,	'lastnamefirstname_comparison_jaro': 0.4555555555555555,	'middlenamefirstname_comparison_jaro': 0.5396825396825397,	'dob_comparison_year': True,	'ssn_comparison_levdam': 0,
'firstname_comparison_jaro': 0.4305555555555555,	'lastnamefirstname_comparison_levdam': 5,	'middlenamefirstname_comparison_levdam': 6,	'dob_comparison_month': True,	'ssn_comparison_exact': True,
'firstname_comparison_levdam': 8,	'lastnamefirstname_comparison_nysiis': False,	'middlenamefirstname_comparison_nysiis': False,	'dob_comparison_day': False,	'ssn_comparison_longestoverlap': 9,
'firstname_comparison_nysiis': False,	'lastnamefirstname_comparison_initialmatch': False,	'middlenamefirstname_comparison_initialmatch': False,	'dob_comparison_monthdayswitch': False,	'ssn_comparison_firstthree': True,
'firstname_comparison_initialmatch': False,	'lastnamefirstname_comparison_contained': False,	'middlenamefirstname_comparison_contained': False,	'dob_comparison_daydiff': 21,	'ssn_comparison_lastfour': True,
'firstname_comparison_contained': False,	'lastnamefirstname_comparison_longestoverlap': 1,	'middlenamefirstname_comparison_longestoverlap': 2,	'sex_eithernull': False,	'ssn_eithernullorinvalid': False,
'firstname_comparison_longestoverlap': 1,	'lastname_eithernull': False,	'middlenamelastname_eithernull': False,	'sex_eitherinvalid': False,	'email_eithernull': False,
'firstnamelastname_eithernull': False,	'lastname_comparison_exact': False,	'middlenamelastname_comparison_exact': True,	'sex_comparison_exact': True,	'email_eitherinvalid': False,
'firstnamelastname_comparison_exact': False,	'lastname_comparison_jaro': 0.5619047619047619,	'middlenamelastname_comparison_jaro': 1.0,	'sex_eithernullorinvalid': False,	'email_comparison_exact': False,
'firstnamelastname_comparison_jaro': 0.6011904761904762,	'lastname_comparison_levdam': 6,	'middlenamelastname_comparison_levdam': 0,	'address_eithernull': False,	'email_comparison_levdam': 5,
'firstnamelastname_comparison_levdam': 6,	'lastname_comparison_nysiis': False,	'middlenamelastname_comparison_nysiis': True,	'address_eitherinvalid': False,	'email_comparison_user_exact': True,
'firstnamelastname_comparison_nysiis': False,	'lastname_comparison_initialmatch': False,	'middlenamelastname_comparison_initialmatch': True,	'address_comparison_street_exact': True,	'email_comparison_user_levdam': 0,
'firstnamelastname_comparison_initialmatch': False,	'lastname_comparison_contained': False,	'middlenamelastname_comparison_contained': True,	'address_comparison_street_jaro': 1.0,	'email_comparison_user_longestoverlap': 23,
'firstnamelastname_comparison_contained': False,	'lastname_comparison_longestoverlap': 1,	'middlenamelastname_comparison_longestoverlap': 7,	'address_comparison_street_longestoverlap': 6,	'email_eithernullorinvalid': False,
'firstnamelastname_comparison_longestoverlap': 1,	'lastnamemiddlename_eithernull': False,	'middlename_eithernull': False,	'address_comparison_number_levdam': 0,	'mrn_eithernull': False,
'firstnamemiddlename_eithernull': False,	'lastnamemiddlename_comparison_exact': False,	'middlename_comparison_exact': False,	'address_eithernullorinvalid': False,	'mrn_comparison_exact': False,
'firstnamemiddlename_comparison_exact': False,	'lastnamemiddlename_comparison_jaro': 0.5407407407407407,	'middlename_comparison_jaro': 0.5886243386243386,	'zip_eithernull': False,	'mrn_comparison_levdam': 5,
'firstnamemiddlename_comparison_jaro': 0.6481481481481481,	'lastnamemiddlename_comparison_levdam': 8,	'middlename_comparison_levdam': 5,	'zip_eitherinvalid': False,	'mrn_comparison_longestoverlap': 7
'firstnamemiddlename_comparison_levdam': 7,	'lastnamemiddlename_comparison_nysiis': False,	'middlename_comparison_nysiis': False,	'zip_comparison_exact': False,	
'firstnamemiddlename_comparison_nysiis': False,	'lastnamemiddlename_comparison_initialmatch': False,	'middlename_comparison_initialmatch': True,	'zip_comparison_levdam': 4,	
'firstnamemiddlename_comparison_initialmatch': False,	'lastnamemiddlename_comparison_contained': False,	'middlename_comparison_contained': False,	'zip_eithernullorinvalid': False,	
'firstnamemiddlename_comparison_contained': False,	'lastnamemiddlename_comparison_longestoverlap': 1,	'middlename_comparison_longestoverlap': 2,	'phone_eithernull': False,	
'firstnamemiddlename_comparison_longestoverlap': 2,		'fullname_comparison_jaro': 0.6835497835497836,	'phone_eitherinvalid': False,	

NMDOH
NEW MEXICO DEPARTMENT OF HEALTH

Investing for tomorrow, delivering today.

Use of Smarty in NM
Address Cleansing and Validation

Kathryn Cruz, NMSIIS Manager
May 25, 2022

Use of Smarty in NM

- Reminder Recall Efforts
- Analysis and update of existing patient addresses
  - Manually on patient demographic screen
  - Automated address validation job
- Rooftop Geocoding

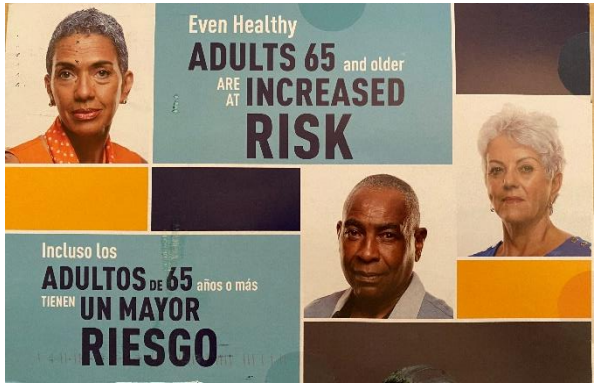
Why Smarty?

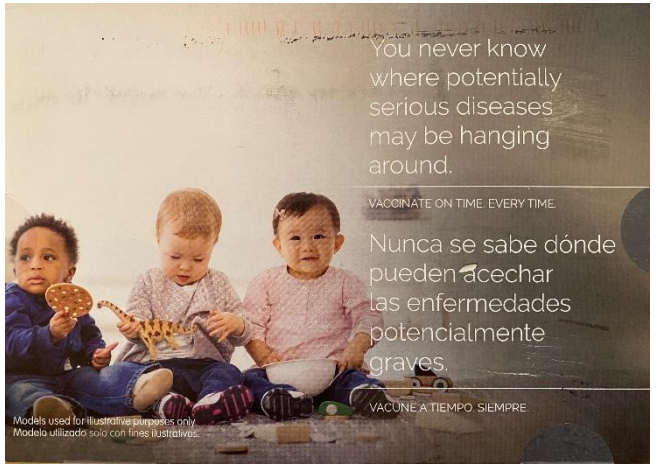
- AIRA Support
- Envision Support
- Set up/implementation
- Confidentiality
- HL7 vs API
- Future plans for data improvement

Reminder Recall Efforts

NM currently uses the Command Line Interface for monthly address data cleansing to include:

- Child well visits
- Child missed dose reminder recall
- Adult reminder recall





Steps to Process a Command Line

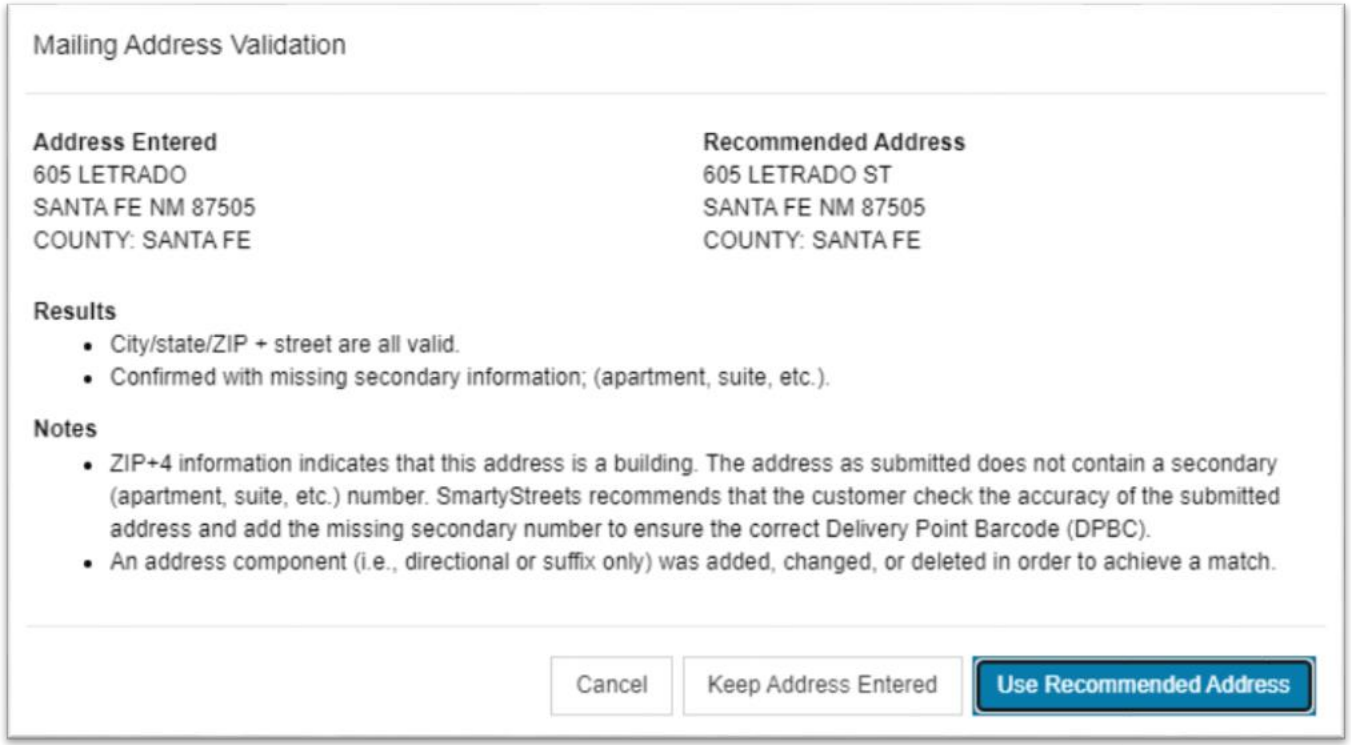
The process/execution of the file is usually less then 30 seconds. The output file exports to a CSV/Excel file.

The output file provides very useful information for the user to review addresses that have a valid address match or non match which could include:

- Match-Vacant
- No Match
- No Match-PO Box Only

Manual Address Processing

Prompted when a user adds or updated a patient address on patient demographic screen

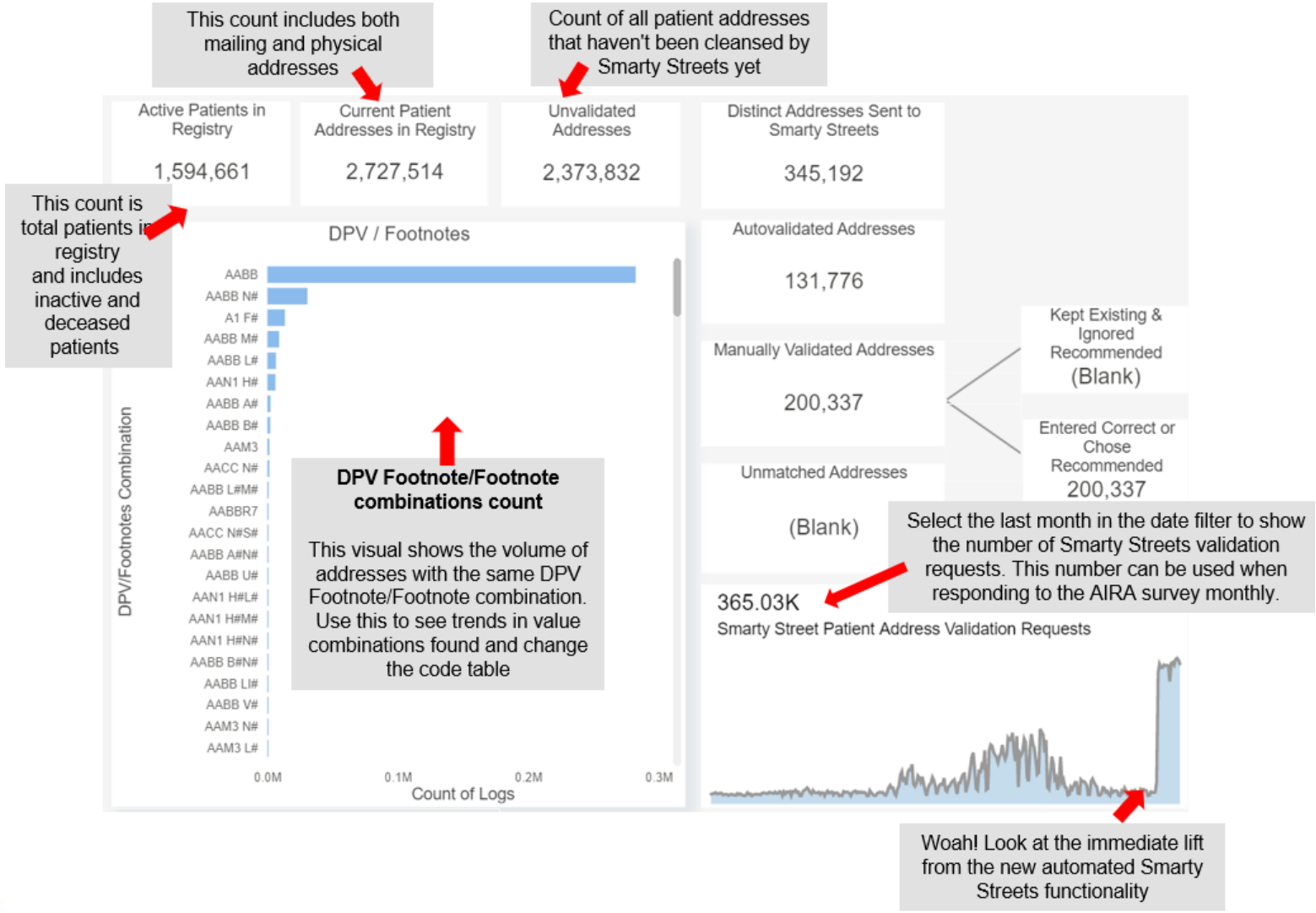


Bulk Address Processing

Existing job within NMSIIS that analyzes existing patient addresses and automatically updates the address, if it meets a certain criteria

- Runs every 15 minutes
- Verifies unvalidated addresses added through HL7 and flat files
- Validation completed on both physical and mailing address

PowerBI Smarty Report



Smarty Rooftop Geocoding

NM participated in the pilot program of using the Smarty Rooftop Geocoding in July 2021

US Rooftop Geocoding indicates the exact location of the structure using latitude and longitude coordinates

Findings of Smarty Usage

- User Friendly
- Fast, simple, efficient
- Saves time and money
- Resources available

- No other departments use Smarty in NM (that we are aware of) but we are advocates for it

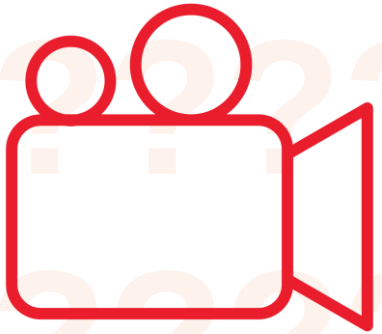
Kathryn Cruz, NMSIIS Manager

Kathryn.Cruz@state.nm.us

Questions and answer

Post in the chat

Raise your hand



Turn on your video

Next Steps

- Post additional questions on Circle - link provided in the chat
- Fifteen minute break 3:30– 3:45 PM EST
- Next session 3:45 PM EST
  - *Workshop reflections*

Thank you.

DMI

Better data. Better decisions. Better health.