

STRATEGIC DATA PROJECT

2026 CONVENING



Building Smart Systems for Better Schools

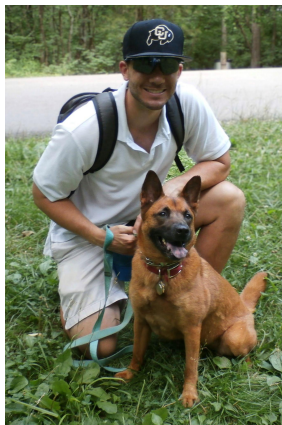
Automation, ML and LLMs to Support Student Success

Arthur Cunningham | Assessment Coordinator, Strategic Data Project Fellow 15 | Metro Nashville Public Schools
Dr. Lief Esbenshade | Assistant Director of Research and Technology Development | University of Washington, AmplifyLearn.AI Center

Introductions

Arthur Cunningham

- **Assessment Coordinator at Metro Nashville Public Schools, Harvard Strategic Data Project Fellow Cohort 15**
- **I've worked in education for 15 years with 10 + being at MNPS within the Research, Assessment & Evaluation Dept.**



Lief Esbenshade

- **Assistant Director AmplifyLearn.AI Center at the University of Washington**
- **10+ years of experience in education data analytics**
- **Worked in edtech (BrightBytes, Google), Schools (NYC, Seattle, Everett CC), and academic settings.**



Agenda for Today

Arthur Cunningham

- Evolution of our data work
- Data Pipeline automation
- Machine Learning
- LLMs that provide targeted support

Lief Esbenshade - Making Complex Analysis Reproducible

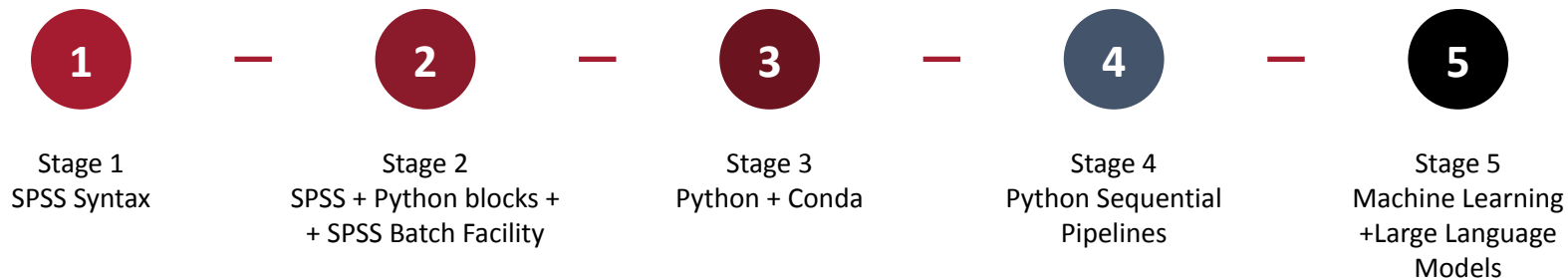
- How we created reports for 12 school districts on Colleague AI Platform Adoption
- Data Pipeline & Standardization
- Visualization Specification & Quarto
- Brief Summary of Report Findings

Questions & Large Group Discussion

Small Group Discussion (5 minutes)

- Turn to the people sitting next to you
- Introduce yourself and how you work with edu data
- Questions to discuss
 - What is the data / technical level at your organization?
 - What are your near term and medium term strategic data priorities for your organization?

Key Objectives



The Journey: Each stage reduced manual effort, increased reliability, and freed staff capacity for higher-value work.

Goal:

Shift staff time from data wrangling to insight on strategic goals, and from reporting to prediction.

Stage 1: SPSS Syntax

Where It All Started

SPSS Syntax Files

SPSS syntax is very useful, but comparatively not very dynamic. Has its own macro type language for building reusable functions with limited documentation and hard to debug.

File Handle System

SPSS file handles define directory paths. Static, hardcoded breaks every year end when folders change. Naming conventions can be an issue.

```
*SPSS MACRO SPLIT CSV.
#####
dataset close all.
get file='c:\datab\2025\active.sav'.
string slabel(a55).
compute slabel=VALUELABEL(meno2).
sort cases by meno.
SAVE OUTFILE='c:\datab\26\splitter\mydata.sav'.
* Now start the job.
set mprint=on.
#####
!DEFINE !SPLIT (var=!TOKENS(1) /value=!TOKENS(1) )
TEMPORARY.
SELECT IF (!var=!value).
SAVE TRANSLATE OUTFILE=!QUOTE(!CONCAT('c:\datab\26\splitter\'!,EVAL(!UNQUOTE(!value)),'.csv'))
/TYPE=CSV
/ENCODING='Locale'
/MAP
/REPLACE
/FIELDNAMES
/CELLS=LABELS.
execute.
!ENDDEFINE.
#####
* Find all different existing values of cat1.
AGGREGATE /OUTFILE=*
/BREAK=slabel /notused = N.
* Write a syntax file which will call the above macro.
WRITE OUTFILE='c:\datab\26\splitter\temp.sps'
/!SPLIT var=slabel value=""slabel"!'.
Execute.
* Get the original data file and do the macro calls.
GET FILE='c:\datab\26\splitter\mydata.sav'.
insert file = 'c:\datab\26\splitter\temp.sps'.
#####
```

Stage 1: SPSS Syntax

INSERT Chains & Manual Execution

INSERT Chains

Large syntax broken into smaller .sps files and chained together using INSERT commands for modularity. Debugging nightmare.

Manual Execution

If a job needs updating, staff member opens syntax file, replaces paths, file names, runs, and either emails or loads file to system.

```
2297 *insert charter clean file & email file.*****.
2298 *dataset close all.
2299 insert file="C:\General Syntax 22-23\SPSS Production\ELL Production Draft.sps".
2300 insert file="C:\General Syntax 22-23\SPSS Production\Accommodations Draft 11.15.22.sps".
2301 insert file="C:\General Syntax 22-23\SPSS Production\TN Check 4.26.22 TCAP Alt Pearson.sps".
2302 insert file="C:\General Syntax 23-24\Clean Charter Folders 6.13.23.sps".
2303 insert file="C:\General Syntax 23-24\Data Quality Notes to Charters 10.4.23.sps".
2304 insert file="C:\General Syntax 23-24\BI Copies 5.4.23 draft.sps".
2305 insert file="C:\General Syntax 23-24\Testing Coordinator Info 6.22.23.sps".
2306 insert file="C:\General Syntax 22-23\SPSS Production\Power BI list 8.25.23.sps".
2307 insert file="C:\General Syntax 23-24\Data Quality automate.sps".
2308
```

Stage 2: SPSS + Python + Batch Facility

Automating Execution Without Changing the Scripts

This stage wouldn't have been possible without the **Mid-Atlantic Regional Educational Laboratory (REL Mid-Atlantic)**
Their technical documentation on IBM SPSS batch facility was a gateway.

IBM SPSS Command Line Reference

Create .BAT files that run in background with Windows Command Interpreter. .SPJ file is SPSS production file with path to SPSS syntax.

```
stats.exe ^
  spss_prod.spj -production silent
```



Stage 2: SPSS + Python + Batch Facility

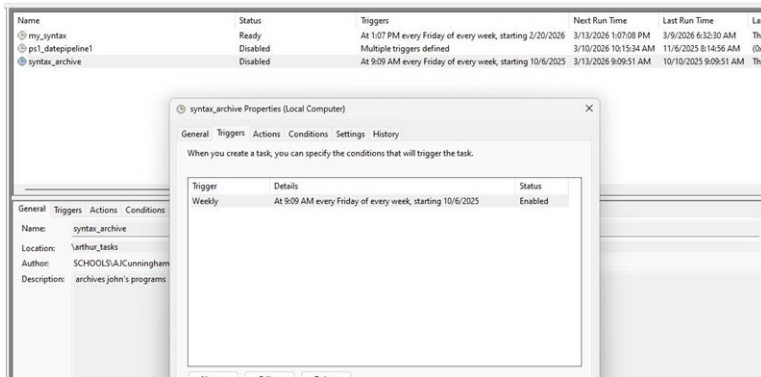
Windows Task Scheduler

- Tasks in named folders per user/project
- Triggers: weekly, daily, at logon, on-demand
- Actions call a .bat or .ps1 to launch SPSS
- Runs silently, no open windows needed

IBM SPSS Extension Hub

Extension Hub hosts both IBM authored extensions and community contributed ones, covering machine learning, visualization, data wrangling, and psychometrics.

Jon Peck, Ph.D.



Stage 3: Python + Conda

Scripts, Environments & Reproducible Workflows

What Python Replaced

- Hard coded file names → globbing wildcards
- Year-specific paths → regex pattern matching
- Manual copy-paste → `shutil.copy2` automation
- One-off scripts → reusable, parameterized jobs

```
improved copy job.py X
7 # pylint: disable=invalid-name
8 import os, shutil, glob
9 from pathlib import Path
10
11 part1 = '//fileshare.mnps.org'
12 part2 = 'rda$'
13 part3 = 'Data'
14 part4 = '202[4-9]'
15 pseudo_path = os.path.join(part1, part2, part3, part4, "py_pipeline_marker.txt")
16 file_path = Path(pseudo_path).parent
17 print(file_path)
18 print(f"Source directory: {file_path}")
19 print(f"Marker file path: {pseudo_path}")
20
21 # Define destination folder
22 dst_folder = r'c:\datab\2026'
23 # Ensure destination directory exists
24 os.makedirs(dst_folder, exist_ok=True)
25
26 #active
27 try:
28     py_pattern = file_path / 'Active**2026*.sav' #will need to change this next year#
29     py_files = glob.glob(str(py_pattern))
30     if not py_files:
31         raise FileNotFoundError(f"No files match {py_pattern}")
32     active_path = max(py_files, key=os.path.getmtime)
33
34     # Copy the most recent file
35     file_name = os.path.basename(active_path)
36     dst_path = os.path.join(dst_folder, file_name)
37     shutil.copy2(active_path, dst_path)
38     print(f"Copied: {active_path} to {dst_path}")
```

Stage 3: Anaconda Environments

Managing Python Environments

Anaconda lets you create isolated environments. Each project gets its own Python version and packages, no conflicts.

```
conda create -n luigi_env python=3.11
conda activate luigi_env
conda install pandas luigi pyreadstat
conda env export > environment.yml
```

Tools & IDEs

PyCharm, VS Code, Jupyter Notebook/Lab, Spyder, Wing IDE, Positron, Idle, Cursor

Text Editors:

Sublime Text, Notepad++, Vim, NeoVim

Package Managers:

pip, conda, uv, poetry, mamba, pixi, virtualenv

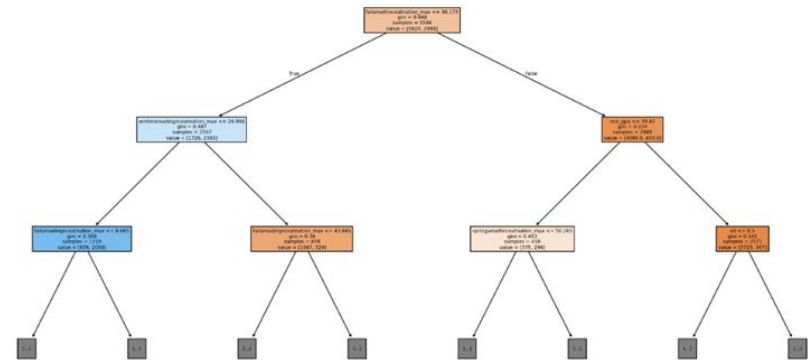
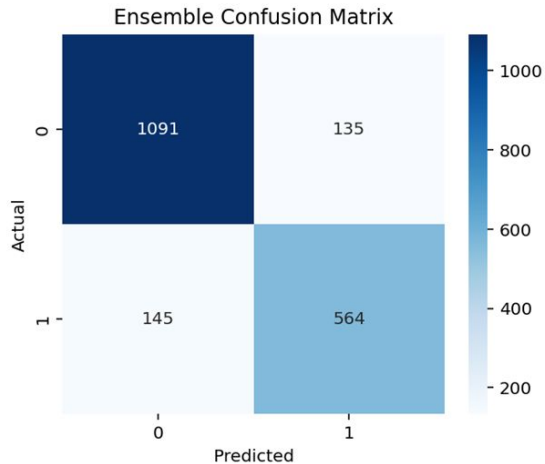
My setup:

miniconda, spyder, idle, notepad++, SPSS

Stage 5: Machine Learning

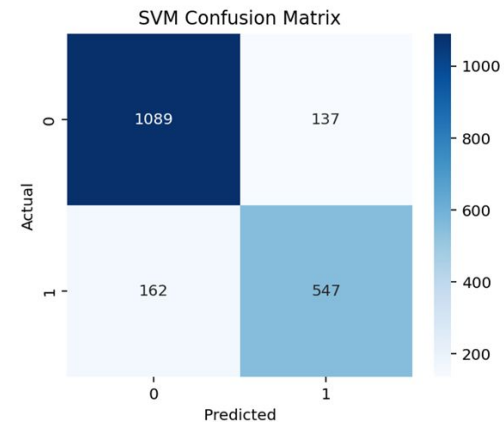
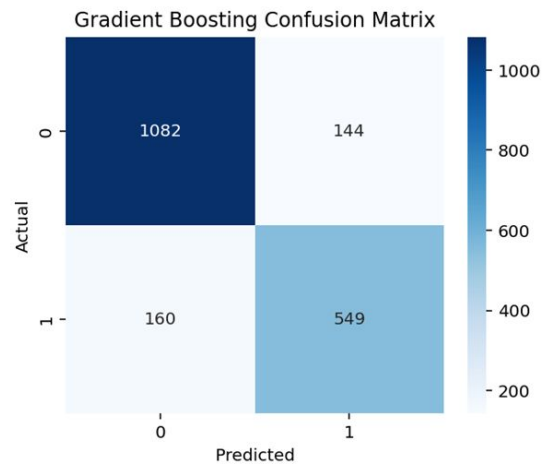
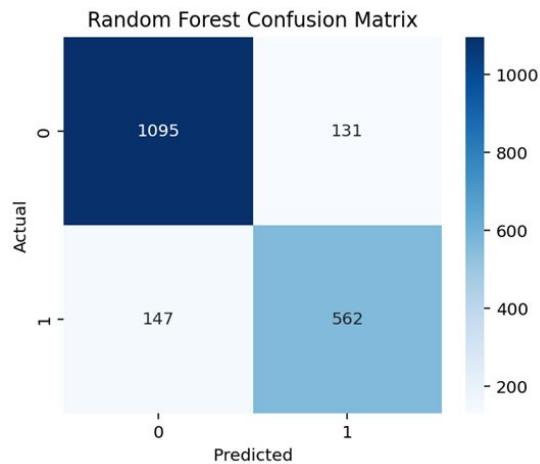
Machine Learning to help predict & support student program interventions

The goal was to provide an early warning system trained on two years of prior data, to help predict which students needed tutoring intervention before the end of 3rd grade. Dr. Lief & Dr. Changas made this possible.



Stage 5: Machine Learning

Machine Learning to help predict & support student program interventions



Stage 5: LLM

LLM chatbot to answer building level leaders assessment related queries.

UW hackweek team created first version of RAG (Retrieval Augmentation Generation), using Ollama + Python + Streamlit



From across the nation, we are **Team (B)RAG**

- **Arthur Cunningham**
 - Nashville, TN
 - Metro Nashville Public Schools
- **Ecy King**
 - Los Angeles, CA
 - McGraw Hill
- **Katelyn Egan**
 - Albany, NY
 - Curriculum Associates
- **Nedim Yel**
 - Boston, MA
 - Data Analysis and Statistical Solutions
- **Nick Teal**
 - Seattle, WA
 - Kent School District

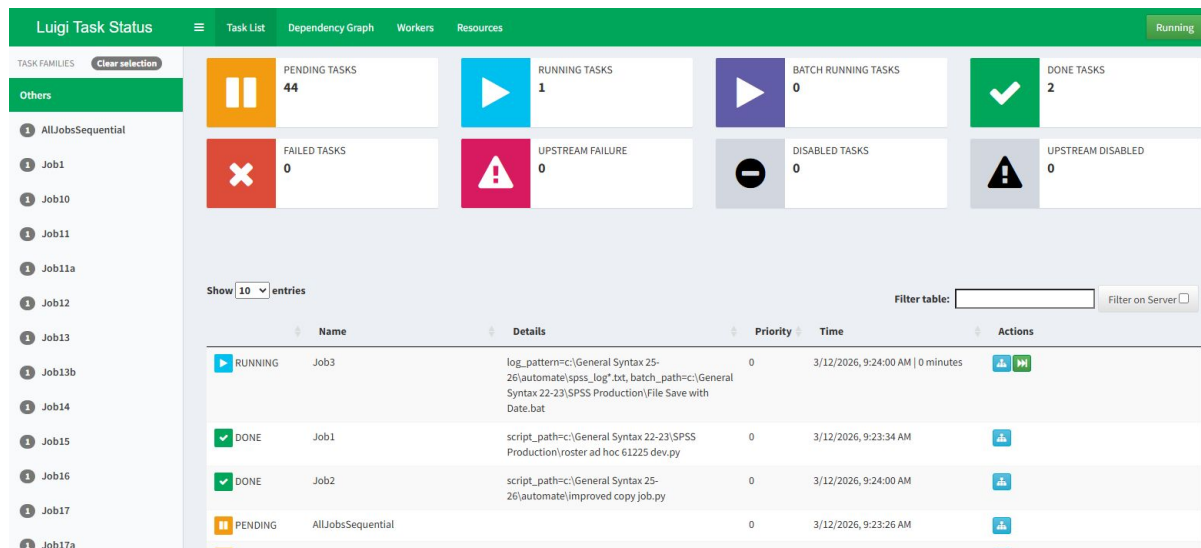
Stage 4: Python Sequential Pipelines

Dependency-Aware Job Chaining; Luigi

Made possible by **ISEA: A Data Science Training Program** to Advance Educational Research and Practice
Min Sun, Ph.D., David A. C. Beck, Ph.D., Lief Esbenshade, Ph.D.

UW College of Education, UW eScience Institute — launched January 2024 via 3-year IES grant

- Job dependencies: Sequential runs
- Auto retry logic, SPSS legacy scripts output checked via Python
- Mix Python scripts and SPSS batch jobs in one pipeline
- Single PowerShell trigger runs jobs hands free



Stage 4: Python Sequential Pipelines

Dependency-Aware Job Chaining; Prefect

Made possible by **ISEA: A Data Science Training Program** to Advance Educational Research and Practice
Min Sun, Ph.D., David A. C. Beck, Ph.D., Lief Esbenshade, Ph.D.

UW College of Education, UW eScience Institute — launched January 2024 via 3-year IES grant



Stage 5: LLM

LLM chatbot to answer building level leaders assessment related queries.

The most recent version uses Anthropic + Python + uvicorn + FastAPI + Railway + Duckdb. Still prototype hopefully will launch this summer.

MNPS Assessment Research Assistant
Metro Nashville Public Schools - AI-Powered Document Search

08:57 AM

How can I change and add sessions within Pearson? You

Creating and Editing Sessions in PearsonAccess Next

You can create and edit sessions individually or in bulk using the export/import tool. Here's how to do it using the bulk method:

To Edit/Create Sessions via Export/Import:

1. Go to **Setup > Import/Export Data**, then select **Import/Export Data** from the Tasks menu and click **Start**
2. From the Type dropdown, select **Session Create/Move Export** and click **Process**
3. Wait for the file — you'll receive an email from Pearson (noreply@mail.pearsonaccessnext.com) or refresh the **View File Details** page every few minutes
4. **Download the file**
5. Go to **Column K** to change session names — all students in the same session must have the **exact same session name**

To Create/Edit Sessions Directly in PAN:

1. Go to **Sessions** in PearsonAccess Next
2. Verify students are in pre-loaded sessions by clicking next to the session name
3. Use **Select Tasks > Create/Start** to create a new session, or select an existing session name to edit it

> **Note:** Pearson will automatically pre-load test sessions from the TCAP Visibility Tool, but BTCs may create custom sessions as needed. All sessions should be created/edited

Ask a question about MNPS assessments...

Ctrl+Enter to send - Responses drawn from uploaded assessment PDFs

Where We're Heading hopefully 😊

Machine Learning + LLMs to Support Student Success

Predictive ML + Modern Reporting

- Early warning for at risk students
- Text analytics and natural language processing
- Interactive data visualizations & web applications
- Programmatically evaluating strategic goals

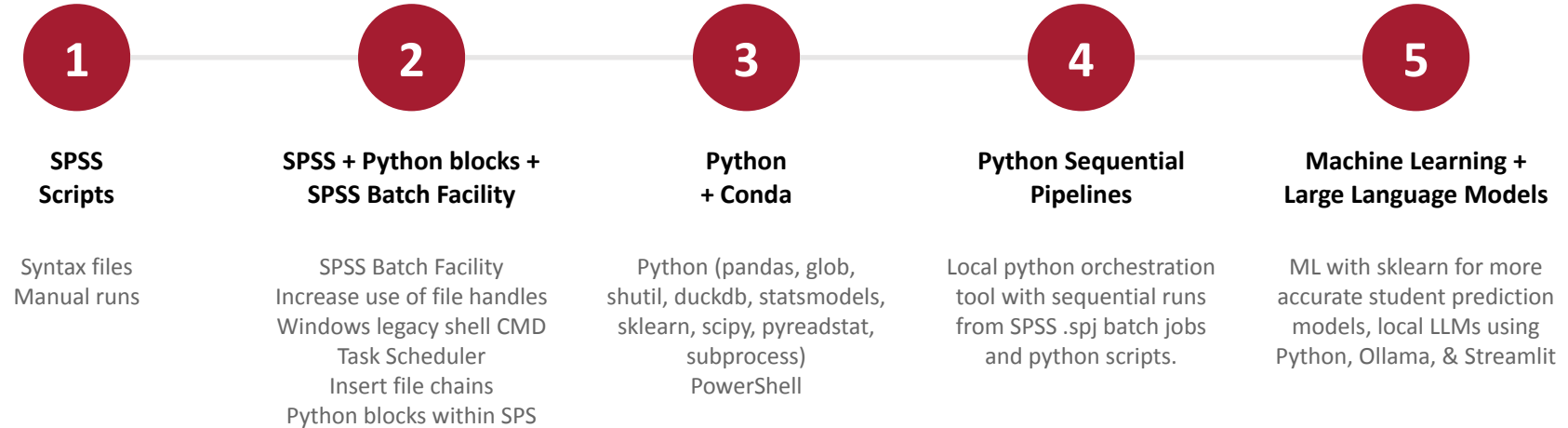
LLM Integration

- Natural language queries and bots trained on local data to answer high volume questions to support staff & students
- LLM generated scripts from prompts based on current codebase
- LLM-assisted data quality checks
- Auto generated narrative reports with next steps for staff & students

Smarter Pipelines + Modern Stack

- Robust pipelines with advanced error handling
- Automated workflow & data documentation
- Specialty agents as assistants
- Cloud based architectures

The Evolution at a Glance



Key Insight: Each stage reduced manual effort, increased reliability, and freed staff capacity for higher-value work.

Promising Tools Worth Trying

Building Toward a More Dynamic Data Stack

Orchestration tool options: Luigi, Prefect, Flyte, Dagster, & Airflow

Pandera

Data schema checks for ingesting files in pipeline. Define column types, value ranges, and nullability. Great at finding soft errors.

Quality gates for pipelines

dlt

Lightweight Python library that turns raw data sources into clean, versioned, warehouse-ready datasets without boilerplate.

Replace manual file copy scripts with typed pipelines

DuckDB

SQL directly over CSV, Parquet, and .sav/.sas7bdat/.dta files. No server needed. Run queries inside Python or CLI. Written in C++ fast.

Great for in memory analytics, fast SQL on parquet files

Playwright

A browser automation framework that lets you programmatically control web browsers,, form filling, navigating, and extracting data programmatically.

Great for automating browser tasks

Ibis

Use the same API for 20+ backends. Fast local data frames with DuckDB (default), Polars, or DataFusion. Iterate locally, deploy remotely.

Open source dataframe library for any data system

Dask

A parallel computing library for Python designed to scale data workflows beyond what fits in memory. Dask breaks large datasets into smaller chunks and processes them in parallel

Good for workflows beyond memory limits and working across computer cores or clusters.

Discuss at Your Tables (5 min)

1

What data challenges do you have where automation could make things easier?

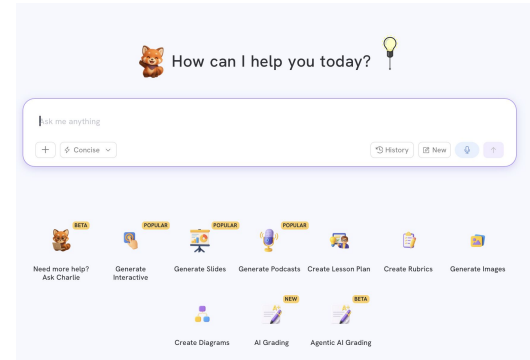
2

What would you need to try implementing them at your organization?

Report Out & Group Conversation (5 min)

AmplifyLearn.AI and Colleague AI

- Federal Grant Funded Research Center at University of Washington
- Research on Generative AI in Education
- Comprehensive AI platform for K-12 education
- Spin off from UW
- Free to use with paid tier



This work is supported by the Institute of Education Sciences of the U.S. Department of Education, through Grant R305C240012 and by several awards from the National Science Foundation (NSF #2043613, 2300291, 2405110) to the University of Washington, and a NSF SBIR/STTR award to Hensun Innovation LLC (#2423365). The opinions expressed are those of the authors and do not represent views of the funders.



Implementation Study

- 2025-26 School Year
- 12 Washington State School Districts
 - ~10K teachers work in these districts, serve ~175K students
- Data Sources
 - Interviews with 20 teachers about AI Use
 - *School District Administrative Data*
 - *Colleague AI Platform Log Data*
- Mid-year Report
- What do we see in how teachers are using Colleague AI from September 1 to December 31 2025



The Goal & The Challenge

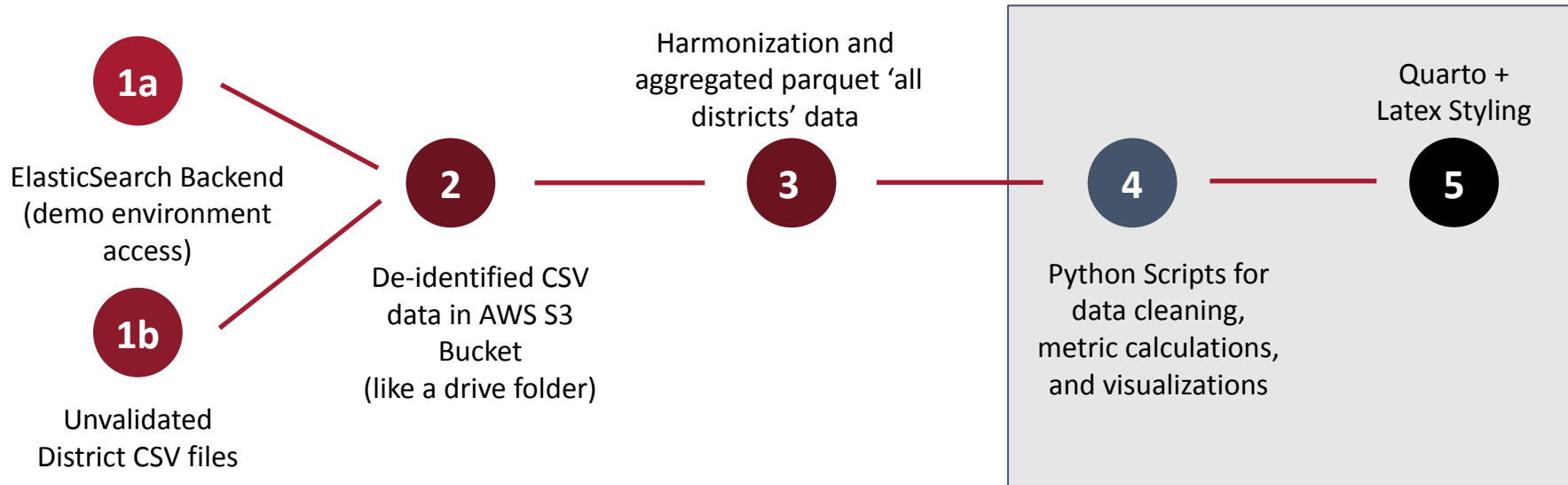
Goal

- Create professional reports for 12 districts & one overall report
- What we have learned about how their teachers are using Colleague AI

Challenge

- Requested administrative data from 12 districts
- Colleague AI data in ElasticSearch (new to me!)
- FERPA and IRB constraints, de-identification before analysis
- Team of 2
 - Me and one grad student

Templated Reporting with Quarto



0 Software Tools

- IDE: integrated development environment: file explorer, variable explorer, plot viewer, data viewer, console and terminal
 - RStudio
 - Positron
 - Spyder
- GIT
 - blame view and edit without risk
- AI tools
 - This space is changing fast, but real potential

IDE: Positron

The screenshot displays the Positron IDE interface with the following components and annotations:

- EXPLORER:** Shows a project structure with folders like 'src' and 'LEGACY_analysis', and files such as 'test.py'. A red arrow points to the file list with the text: "See every file in the project, easily search through them all".
- Code Editor:** Displays Python code for data analysis, including loading data, filtering, and plotting. A red arrow points to the code with the text: "Can run code interactively to the console".
- SESSION VIEWER:** Located at the top right, it shows loaded data objects like 'sessions' and 'trimmed'. A red arrow points to it with the text: "Session viewer, can see all the data i've loaded".
- CONSOLE/TERMINAL:** At the bottom, it shows the execution environment and terminal output. A red arrow points to it with the text: "Can use the terminal to run scripts or create quarto reports".
- PLOTS:** A histogram plot is visible in the bottom right corner, showing the distribution of session lengths. A red arrow points to it with the text: "See plots".

1 Raw Data -> 2 De-identified Data

A) Elasticsearch

- Colleague AI granted access to a test environment that had the data schema
- Over 100 'indexes' had to sift through them and figure out what we wanted to pull
- Used AI tools to write query syntax for desired data

B) District Data Files

- CSV upload portal
- Documented desired schema, but intentionally not enforced with validation on upload
- Stored securely by Colleague AI with infrastructure for platform PII

Email as Join Key - Hash with Pepper

- Hashing prevents going backward, can't turn a hash back into an email
- 'Pepper' adds a random secret to the email and then encrypts it, so i can't look up the hash for `lief@uw.edu`
- De-identified data files shared with research team in AWS S3 in per-district folders

Step 3 Harmonization...

CONFIG file - one place to update

```
S3_BUCKET = "research"
```

```
ORGANIZATIONS = [  
    "abc",  
    "def" ]
```

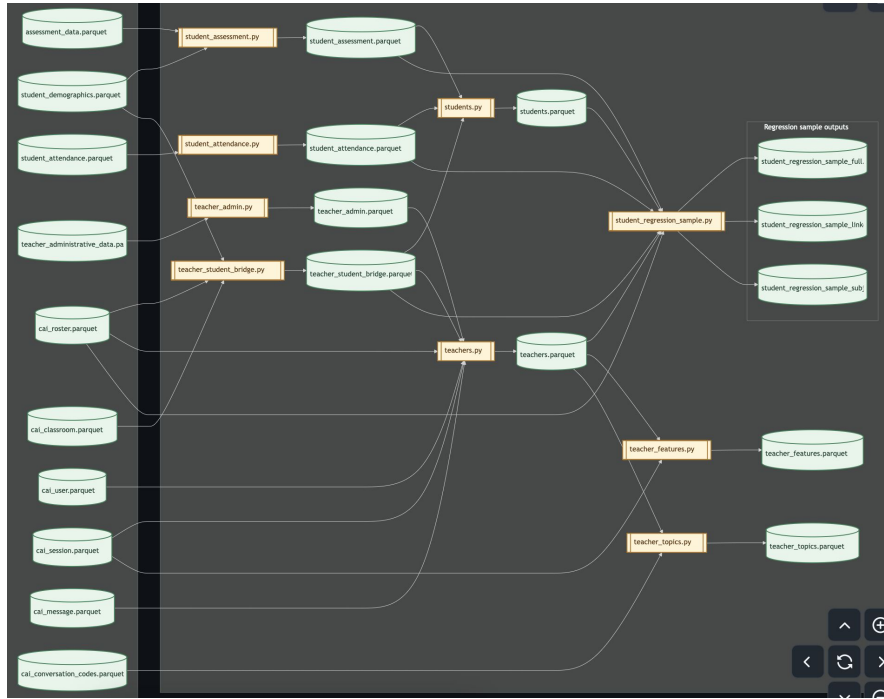
YAML for field renaming - 1 file per district:
{schoolDistrict}.yaml

```
student_demographics:  
  filename: "studentDemographics.csv"  
  columns:  
    email: "email"  
    grade_level: "grade_level"  
    student_with_disability: "student_with_diability"
```

... and Aggregation

- Loop through Organizations, append the data together and store All_District Parquet files. Approximately 200 mb.

Step 4.A Data Cleaning & Processing: *A pretty knot is a safe knot*



I used to make these diagrams by hand, now working on using LLMs to process my scripts and keep them up to date.

\$5 words

- Checkpointing
- DAG - 'Directed Acyclic Graph'
- Idempotent (append vs overwrite)

Details Here

- 10 Input tables
- Raw tables should only be cleaned once
- Regression analysis table construction is much more difficult than running regressions!

One Script -> One Output Table

- Never modify your raw data files
 - raw file > Script > clean file
- Each table must have one script that creates it.
 - Horrifying code I've seen:
 - `get_raw.py -> students.csv -> clean.py -> [overwrites] students.csv`
- Each script should create only one table
 - Sometimes there are exceptions, like if you are creating closely related versions of a table (e.g. the different regression sample tables).
- Name the script and the table the same!
 - Consistent verb prefixes:
 - `get_roster.py > roster.csv` {get_ for fetching raw data from a source}
 - `clean_teacher.py > teacher.parquet` {clean for cleaning}

Declare Variables and Grain

One row per (studentEmail, district)

```
OUTPUT_COLUMNS = [  
    "studentEmail", "district", "gradeLevel",  
    "englishLanguageLearner", "studentWithDisability",  
]  
UNIQUE_COLUMNS = ["studentEmail", "district"]
```

```
{... analysis code here ...}
```

```
dup = df.duplicated(subset=UNIQUE_COLUMNS).sum()  
assert dup == 0, f"{dup}duplicate rows on {UNIQUE_COLUMNS}"  
students = df[OUTPUT_COLUMNS]
```

CMD+SHIFT+F in your IDE lets you search through all the code files in your project

Consider whether the **UNIQUE_COLUMNS** check should be an error or a warning

Your data files won't be in git, but this way you know if the columns change.

The work is iterative - the code can't be

- Learn about the data
 - Console, interactive processing
- Reproducible Scripts
 - Terminal, `python clean_teacher.py`

Reproducible Code

- Idempotent
 - running a script repeatedly does not change the output.
- Acyclic
 - Each step depends only on earlier steps
 - Document your dependency graph
- Checkpoint
 - Clean data once and save the cleaned file
- Regularly 'start over', delete cleaned data and run your scripts

4.B Data Visualization with Atomic Graphs and Config

- **Small, reusable chart functions**
 - eg. given a data frame with a group column, a label column, and value column, create the styled dot plot.
 - Creating a wrapper around the existing plot fn so you can call it with one line for different variables.
- **CONFIG for grouping variables**
 - Every plot using the variable is consistent
 - AND it's trivial to create the same plot with a different grouping variable

```
EXPERIENCE_CONFIG = {  
  'bins': {  
    'new': (0, 4),  
    'mid': (5, 10),  
    'experienced': (11, float('inf'))},  
  'labels': {  
    'new': 'Early-Career (0-4 years)',  
    'mid': 'Mid-Career (5-10 years)',  
    'experienced': 'Experienced (11+ years)'},  
  'colors': {  
    'new': '#F4A261',  
    'mid': '#E76F51',  
    'experienced': '#264653'},  
  'order': ['new', 'mid', 'experienced']}
```

5 Quarto & Report Rendering (<https://quarto.org/>)

- **Front matter**
 - **Declare format, lots of supported options including doc, pdf, ppt**
- **This can run through pandoc and apply latex styling**
 - **I used screenshots of previous reports that our designer had made and claude code wrote the latex to implement that styling**

```
---  
title: "Colleague AI Implementation Study:"  
format:  
  pdf:  
    toc: false  
    number-sections: false  
    geometry: margin=1in  
    include-in-header: ../report-assets/report-style.tex  
fig-cap-location: top  
execute:  
  echo: false  
params:  
  organization: "DEMOSCHOOLDISTRICT"  
---
```

5 Quarto

- code blocks (R, Python, others)

```
```\{python}
```

```
...
```

- Markdown text
- `compose_feature_bars` is a standard python function that returns a plot. Quarto automatically renders it

```
```\{python}
teachers =
pd.read_parquet("../data-clean/teachers.parquet" )
teachers = teachers[teachers["district"] == organization]
...
```

```
# Teacher Usage Levels
```

```
## Usage of Colleague AI Increased Steadily Over the Fall
```

```
\{python} f"{n_with:,}"` teachers in your district used
Colleague AI as of December 31, 2025.
```

```
### Feature Usage Overview
```

```
```\{python}
fig = viz.features.compose_feature_bars (features, n_active,
**fig_dims ["feature_bars"])
plt.show ()
...
```

# 5 Quarto & The Command Line

```
quarto render report.qmd -P district:"DIST_NAME"
```

- **Once you've created the report its trivial to run it.**
- **Automatically created 12 beautifully formatted reports, one for each district.**
  - **I rendered to PDF so I couldn't post edit**
  - **Also possible to render to a word doc**

# 5 Quarto Opinions

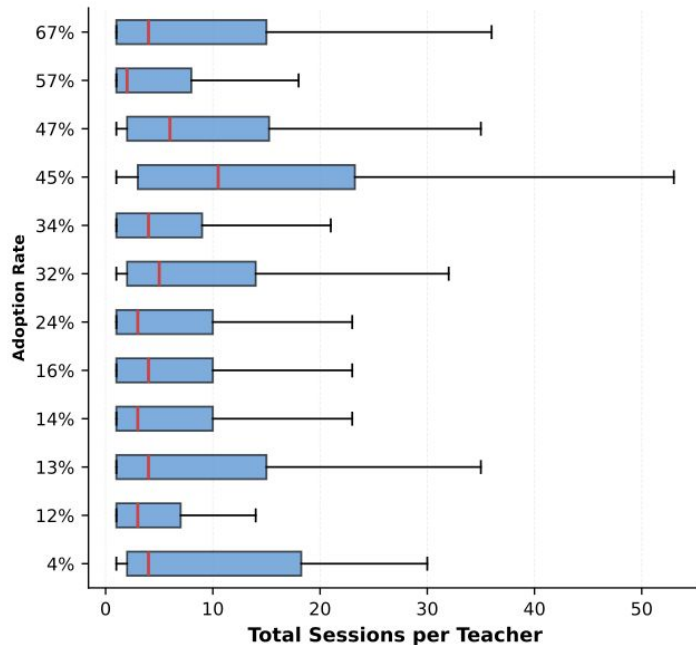
- Don't treat quarto like jupyter notebooks
  - For interactive work, use Positron's built in data explorer and the console.
  - I put everything in scripts that I import, code blocks are just subsetting the data and rendering the plot or printing values.
- Editing the text content of the quarto document was painful.
  - Working on the copy of the report and revising it, it was hard to include other people because we didn't have google docs track changes / commenting, and going through github pull requests was too clunky.
  - Quarto 2 (expected late 2026) will be addressing this

# Coding Recommendations Summary

- Draw a map
- One script per table
- Small set of clean tables
- Create standardized plot & analysis functions
  - with column name parameters
- Quarto
  - filter / load district specific data
  - latex styling to make a PDF report

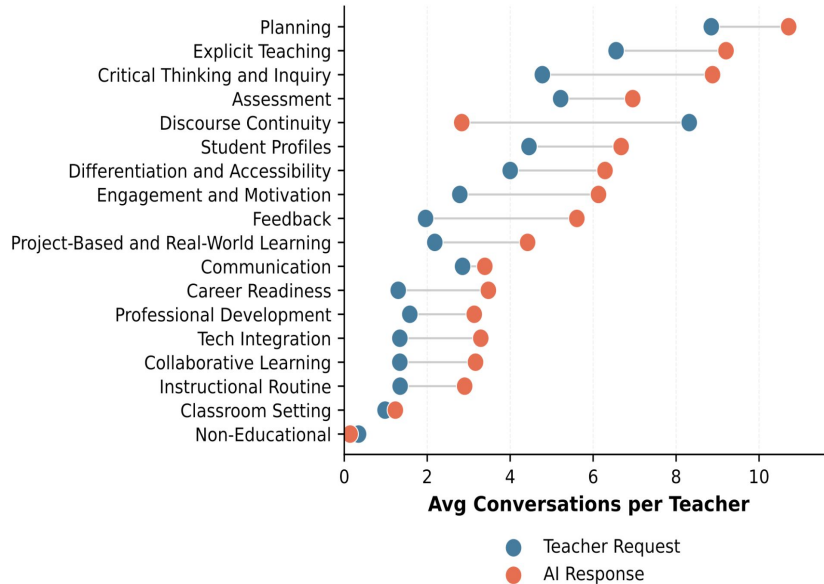
# District Adoption Varied - Extensive and Intensive

Sessions per Teacher by District



- Extensive
  - District adoption varied from 4% up to 67% of teachers trying the platform
- Intensive
  - Overall median teacher had 4 sessions, one district median was 11 sessions.
  - Leadership support, Colleague AI training sessions, and internal training capacity development
- Chat tool “Brainstorm Ideas” made up over 85% of platform usage

# Topics in Teacher-AI Conversations

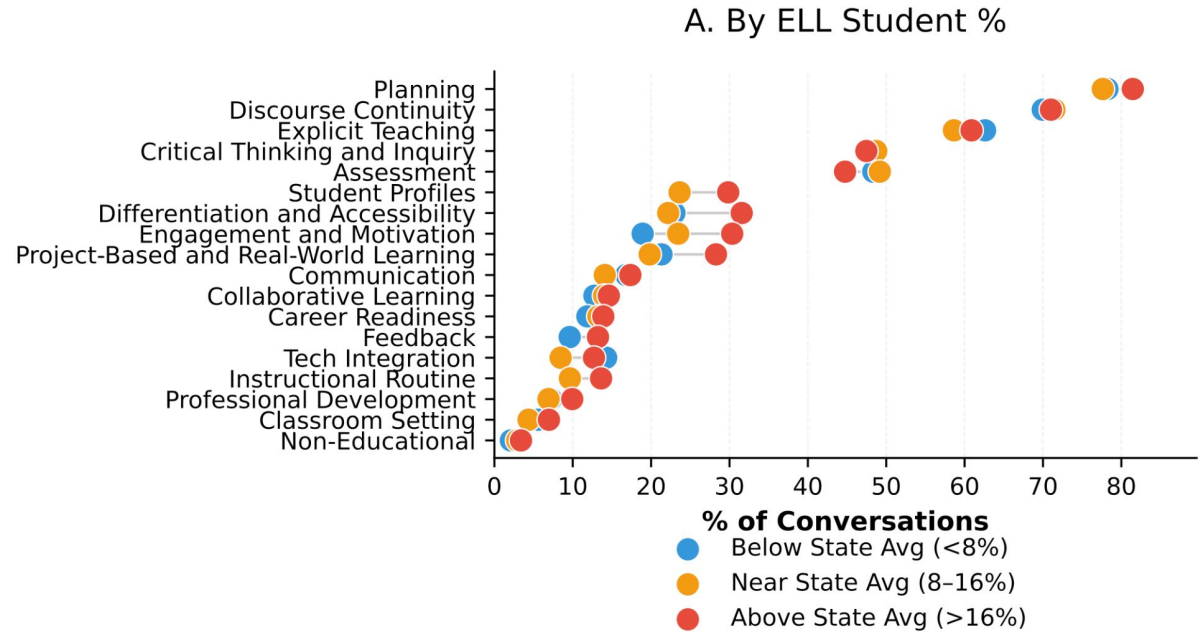


- Colleague AI platform automatically codes topics in teacher-ai conversations
- Almost all conversations are on educational topics
- The AI assistant consistently adds pedagogical topics to the conversations
- Discourse Continuity refers to messages that build on and expand prior messages

For more details on coding of conversation topics, see Liu, A., Esbenshade, L., Sarkar, S., Tian, V., Zhang, Z., He, K., & Sun, M. (2025). How K-12 Educators Use AI: LLM-Assisted Qualitative Analysis at Scale. arXiv preprint arXiv:2507.17985

# Topics in Teacher-AI Conversations

- Teachers with more ELL students send more messages relating to differentiation and engagement
- We see similar patterns for:
  - teachers serving more special education students
  - teachers with higher variance in their students standardized test scores



# Next Steps for AmplifyLearn.AI

- Implementation Study is ongoing, as of March 31st, 61% of teachers in the participating districts have used Colleague AI
- We will be releasing a full year implementation report this summer
- Other work includes
  - ISEA Data Science Training Program Hackweek this summer
  - National survey of Math & Science teacher GenAI use
  - EPP Collaboratory project developing guidance for using AI in teacher training programs
  - We will be offering small research seed grants for early career academics
- Continued R&D with Colleague AI

# Discuss at Your Tables (5 min)

1

Could you apply templated report generating to your own work?  
How would your organization benefit from reproducible templated reports?

2

What's one thing you heard about today that you are interested in trying in the next 6 months?

# Report Out & Group Conversation (5 min)



STRATEGIC DATA PROJECT

2026 CONVENING



# Thank You

Lief Esbenschade [lief@uw.edu](mailto:lief@uw.edu)

[AmplifyLearn.AI](https://AmplifyLearn.AI)

Arthur Cunningham

MNPS | Research, Assessment & Evaluation