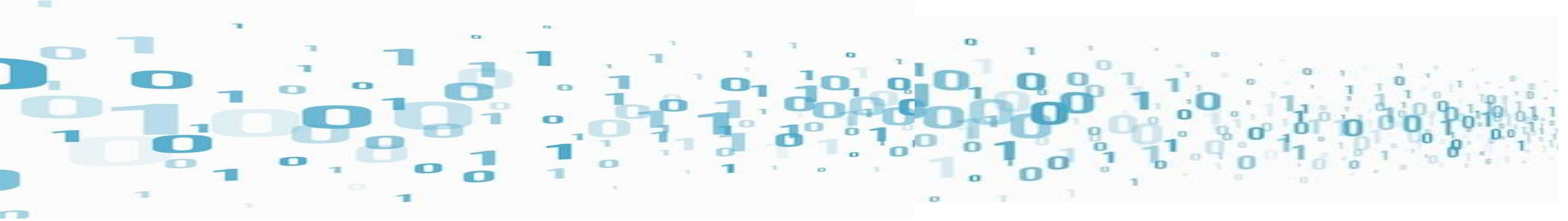
AUTOMATING OCCUPATION CODING (ISCO) IN ZIMBABWE

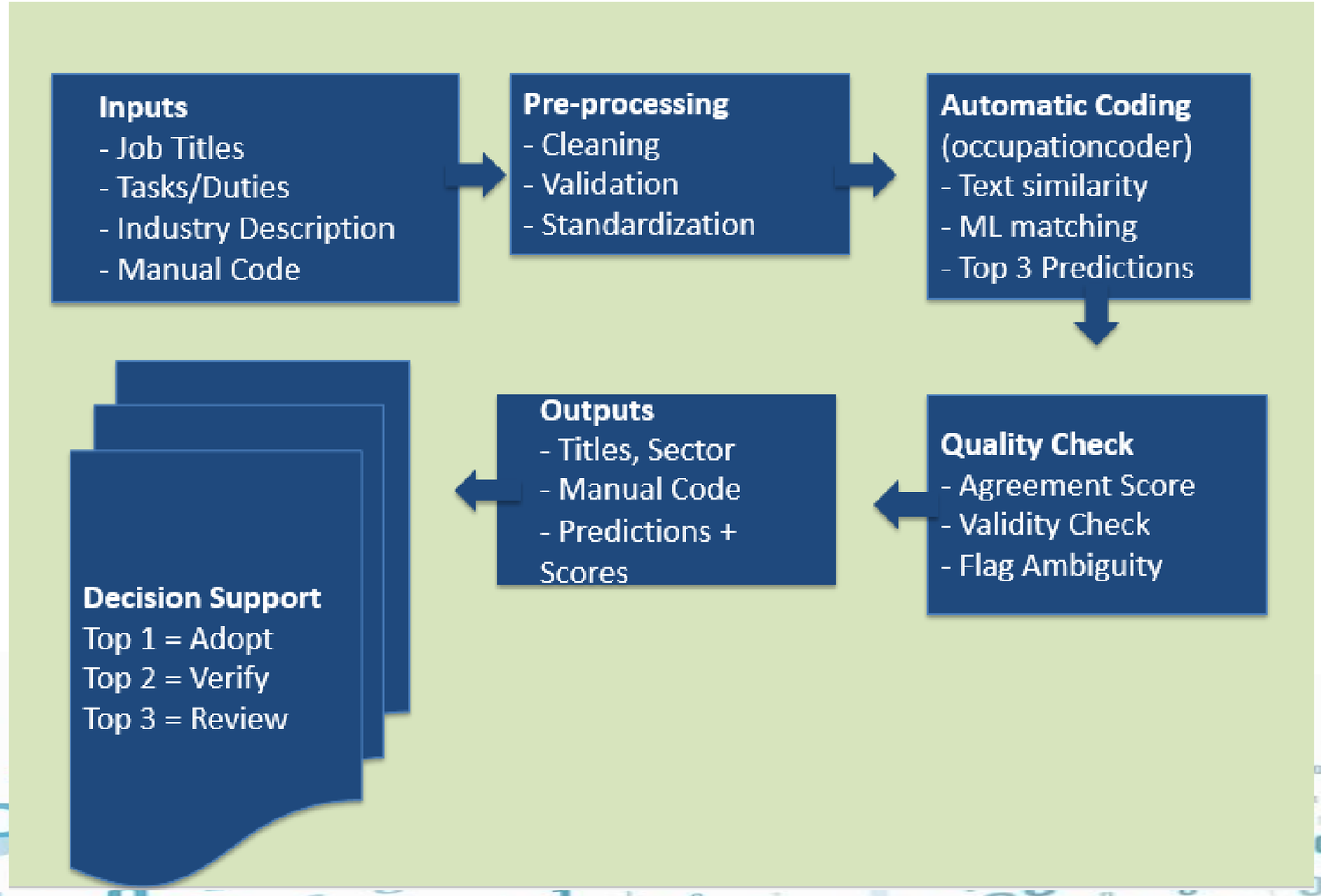
Presented by: CLAPTON MUNONGERWA

Acknowledgements

This tool was developed with technical assistance from UK Office of National Statistics (ONS-UK)



Flow Diagram



Context

- Occupational coding is critical for labour statistics and comparability
- Currently at ZIMSTAT enumerators code occupations manually and verified at data cleaning stage
- Traditionally: Manual verification → time-consuming, costly, inconsistent.
- Solution: Automatic coding using occupation coder (NLP + ISCO mapping)

Inputs

Inputs
- Job Titles
- Tasks/Duties
- Industry Description
- Manual Code

- Input file: XLSX or CSV (stored in /data).
- Required: Job Titles, Manual Codes (ISCO-).
- Optional: Tasks/Duties, Industry Description, Unique ID

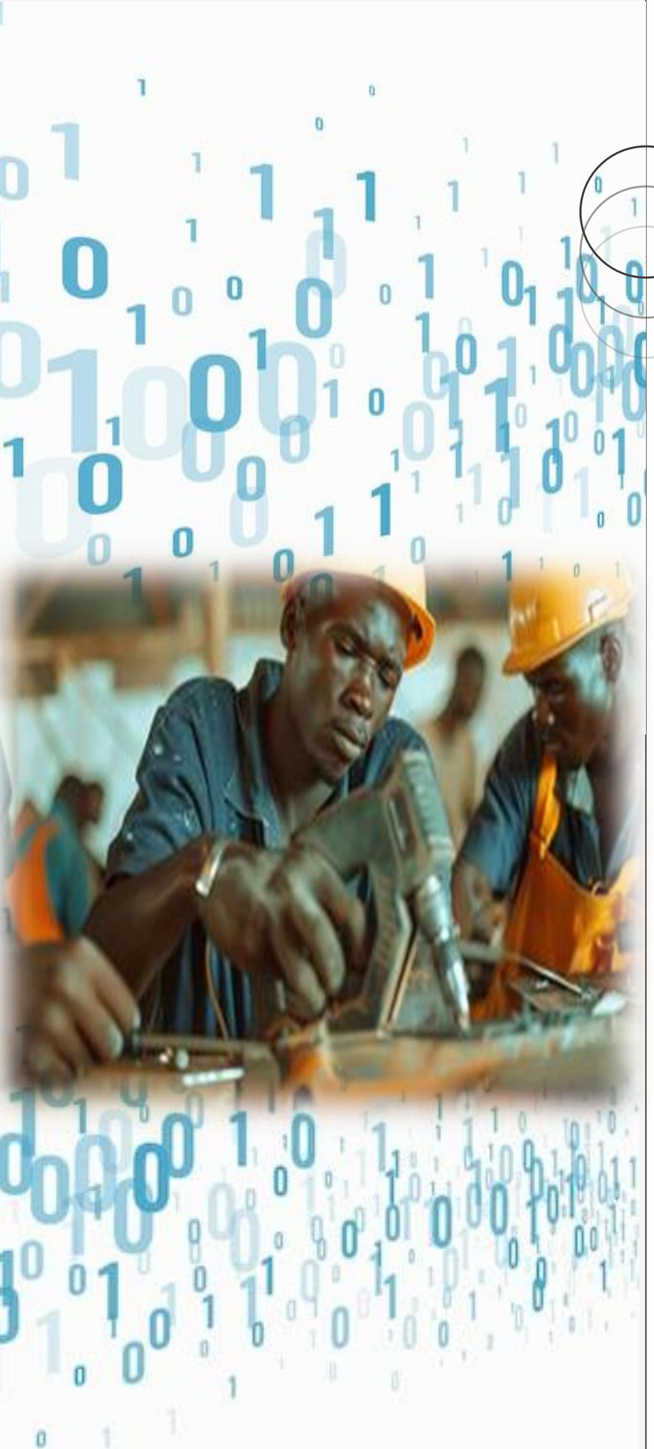
Pre-processing

Pre-processing
- Cleaning
- Validation
- Standardization

- Check uniqueness of column names.
- Rename fields to match occupationcoder requirements.
- Handle missing values (tasks/industry optional).
- Standardize ISCO codes (length, format).

	A	B	C	D	E
1	id	job_title	job_description	job_sector	manual_code
2	1	FARM LABOURER	FARM LABOURER	TOBACCO FARMING	9211
3	2	POULTRY FARMER	RAISE AND SELL POULTRY	RAISING AND SELLING OF POU	6122
4	3	LSC FARM WORKER	REAPING TOBACCO	MAIZE..WHEAT... TOBACCO..S	9211
5	4	CHAINSAW OPERATOR	CUTTING TREES	FORESTRY PLANTATIONS TIMI	8341
6	5	TOBACCO FARMER	GROW AND SELL TOBACCO	CROP AND LIVESTOCK PRODU	6130
7	9	FARM LABOURER	GROWING TOBACCO	TOBACCO	9211

CLAPTON MUNONGERWA



OCTOBER 2025

Automatic Coding
(occupationcoder)
- Text similarity
- ML matching
- Top 3 Predictions

Automatic Coding with occupationcoder

- Core tool: SOCCoder from occupationcoder package.
- Uses text similarity & Machine Learning to map job titles/descriptions → ISCO codes.
- Generates Top 3 predictions for each occupation.
- Provides confidence scores for each prediction.

Core approach

- String similarity and embeddings -> comparing free text job titles, tasks and industry descriptions against standard ISCO descriptions
- SOCCoder component applies a machine learning classifier trained on labelled occupation data

CLAPTON MUNONGERWA

Workflow Interface (Notebook)

- Specify input file & sheet name.
- Define column names (job_titles, manual_codes, etc.).
- Choose coding scheme: isco.
- Run: workflow.workflow(...).
- Output saved in /outputs as Excel file.

Outputs

Outputs
- Titles, Sector
- Manual Code
- Predictions + Scores

- Original job title, tasks, sector.
- Enumerator’s manual code.
- Predicted ISCO codes (Top 1, 2, 3) + Titles.
- Confidence scores (optional).
- Agreement scores exported separately.

Quality Checks

Quality Check
- Agreement Score
- Validity Check
- Flag Ambiguity

- Agreement scores: percent where predictions match manual code.
- Exact matches vs partial matches (Top 2 or Top 3).
- Function code_not_found_in_scheme → flags invalid codes.
- Supports manual verification & feedback loop.

Interpretation of Scores

- Top 1: High confidence, close to perfect match → may be adopted as is.
- Top 2: Moderate confidence → requires verification.
- Top 3: Low confidence → requires detailed review.
- Confidence scores quantify certainty for each prediction.

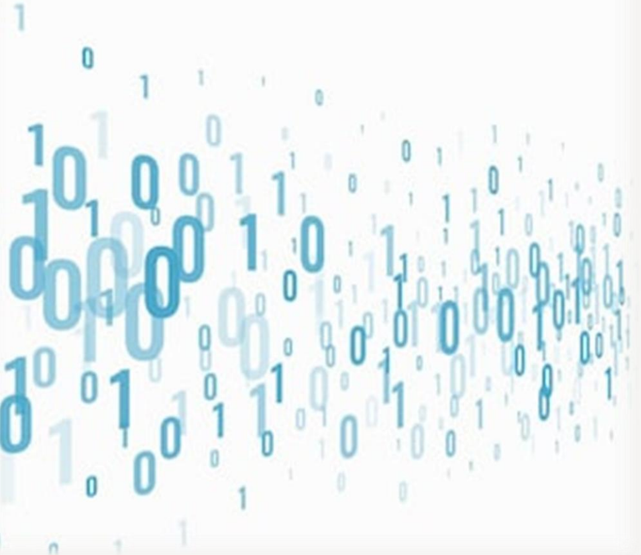
	Prediction	Agreement	Agreement (inc. missing exact)
0	Top 1	0.379321	0.379321
1	Top 2	0.461668	0.461668
2	Top 3	0.489630	0.489630

Advantages

- Efficiency: Faster processing of large datasets.
- Consistency: Reduces subjectivity of manual coding.
- Scalability: Handles large survey datasets.
- Transparency: Predictions with confidence scores aid decisions.

Weaknesses

- Only codes in english though can tolerate near spelling misses
- Dependent on input text quality (typos, vague job titles).
- Language/domain limitations (local terms).
- Human validation required for ambiguous cases.
- Initial setup requires coding scheme dictionaries.



Example

- Enumerator input: “Maize farm worker”
- Manual code: 9211 (Crop farm labourers)
- Predictions:
- Top 1: 9211 (✓ Match)
- Top 2: 6111 (Field crop growers)
- Top 3: 9212 (Livestock farm labourers)
- Confidence: 0.85, 0.72, 0.60

CLAPTON MUNONGERWA

OCTOBER 2025

Conclusion

- occupationcoder package is a practical tool for labour statistics.
- Enhances speed and quality of coding processes.
- Best used as human-in-the-loop system.
- Future: multilingual support.

Links

https://github.com/ZimStat-DataScience/zimstats_industry_occupation_classification

munongerwac@zimstat.co.zw

To recap the highlights:

- ZIMSTAT has tackled the long-standing challenge of manual occupation coding, which was time-intensive and prone to inconsistencies.
- Their solution—the *occupationcoder* tool—uses Natural Language Processing and machine learning to generate ISCO code predictions based on job titles, tasks, and industry descriptions.
- The system produces the top three predictions with confidence scores, enabling a tiered decision-making process: Top 1 predictions may be adopted, Top 2 verified, and Top 3 reviewed.
- This approach enhances efficiency, consistency, and transparency, while still preserving human oversight for ambiguous cases.
- Though currently limited to English and sensitive to input quality, the tool lays a strong foundation for multilingual expansion and regional scalability.