Quality Control of Machine Learning Coding: A Statistics Canada Experience

Javier Oyarzun and Laura Wile (Statistics Canada)

UNECE - ML Group Monthly Forum

October 26, 2022

Delivering insight through data for a better Canada

Outline

- Introduction
- Quality Control (QC) and Machine Learning (ML) at Statistics Canada
- Importance of QC for ML
- QC Methods for ML
- QC Results for ML
- Conclusion

Introduction

- Machine Learning (ML) plays an important role in the mandate of Statistics Canada.
  - The Census of Population and multiple surveys (Labour Force Survey, Statistical Business Register and many others) have started to use fastText (ML method) to code important information.
- ML coding has the advantage to provide accurate, timely and coherent codes for a fraction of the cost.
- As ML is quickly taking a larger role, it’s of primordial importance to quantify the quality of the products/codes that it’s delivering.
- Statistics Canada are actively working to control and assure the quality of their ML products through the help of quality control methodologies.

Evolution of Coding at Statistics Canada

Before 2000s

- Coding:
  - 100% manual
- QC:
  - QC human coders with AOQL approach.

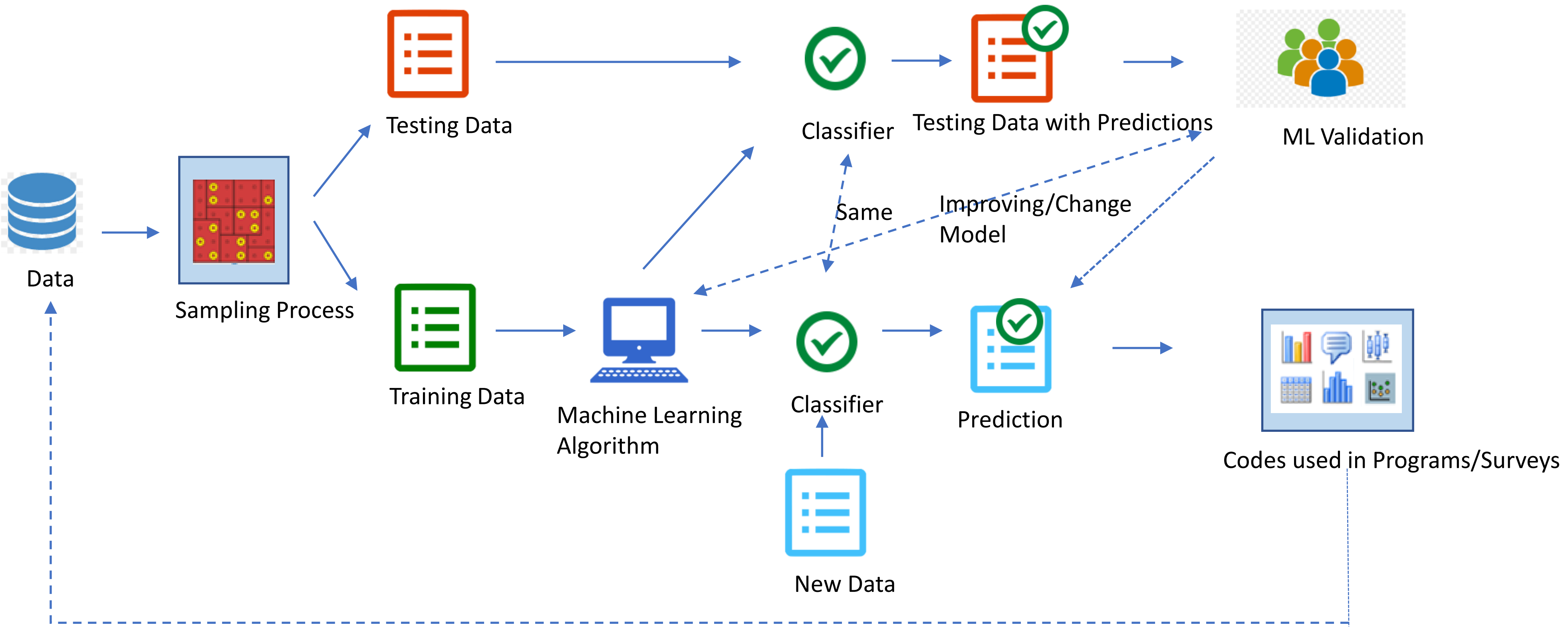
2000 to 2019

- Coding:
  - Mostly manual
  - Automated coding using coding databases
- QC:
  - QC human coders with AOQL and Simple Random Sample (SRS) approaches.

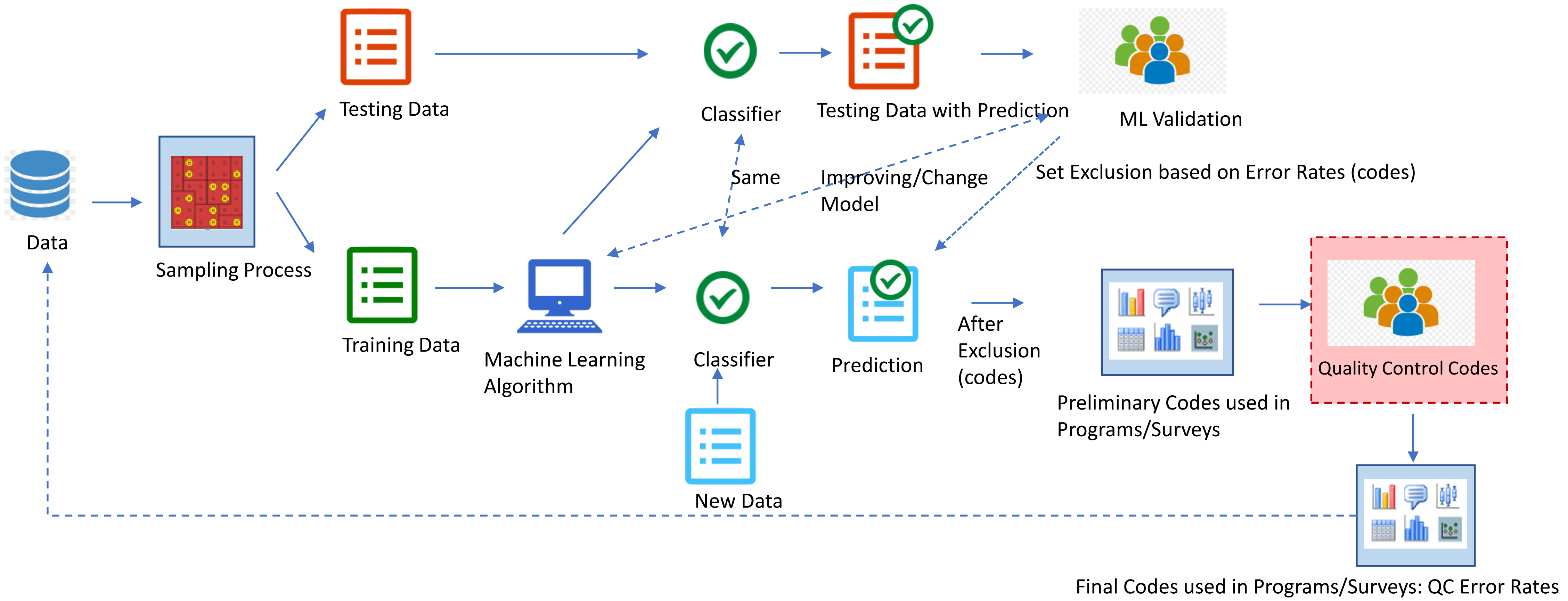
2020 to Present

- Coding:
  - Less manual coding than in the past
  - Automated coding using coding databases
  - ML models
- QC:
  - QC human and ML coders using an SRS approach.

Machine Learning Process (Usual)



Machine Learning Process (Labour Force Survey and others)



Importance of QC for ML Coding

- As with any other models, ML models need to be revised with time, mainly due to “Model Drift”.
  - **Model Drift:**
    - How relationship between the target variable and the independent variables changes with time.
    - Can cause the model to become unstable and the predictions become more erroneous with time (introduction of new classifications, new behaviours, etc.).
    - <u>Solution</u>: Retrain ML models!

Importance of QC for ML Coding

- Without **quality assurance (QA)** or **quality control (QC),** there is no indication of when retraining should be performed.
  - Although some survey programs may choose to retrain models on a regular basis.
  - What data should be used to retrain the ML model?
    - Data that has been running with ML for years?
    - Only data that has been verified?
    - Only data that has been manually coded?

Importance of QC for ML Coding

- **Quality Assurance (QA)**
  - Identifies issues with the model (model drift or other issues, e.g., processing issues).
- **Quality Control (QC)**
  - Produces estimates to track prediction error rates (can also ensure a minimum quality requirement).
  - QC can also provide data that has been vetted/verified in order to retrain future ML models.
  - Corrects data that was wrongly coded by the ML process.

QA and QC of ML at Statistics Canada

Tier 1 (Bronze): QA of testing/validation data

- Coding rate, model precision/recall (based on test/validation data), F1-Score, etc.

Tier 2 (Silver): QC process using optimized SRS to reduce outgoing error rate and human coding workload

- Calculates the incoming and outgoing error rate of ML process
- QC sampling rate: anywhere from 1% to 50% (sometimes at 100%)
- Can approximate workload

Tier 3 (Gold): QC process using Acceptance Sampling

- Ensured outgoing error rate of ML
- QC sampling rate: anywhere from 1% to 100%
- Can approximate workload (more difficult to estimate)

Tier 3 (Gold): Acceptance Sampling (QC)

- **Acceptance Sampling** is a quality control technique that establishes the sample design and the decision rules to determine which batches are acceptable or unacceptable.
  - In its simplest form, acceptance sampling divides the work into batches, selects and checks a sample from each batch and then accepts/rejects the batch depending on the number of errors in the sample.

- The **Average Outgoing Quality Limit (AOQL)** is an acceptance sampling methodology that ensures the overall quality of the work is above a certain quality level (or outgoing error rate is below a predefined level).
  - The cost of AOQL can grow significantly if the incoming error is much larger than the desired outgoing error.

Tier 2 (Silver): SRS QC

- **Quality Control using a Simple Random Sample (SRS)**
  - Correct a proportion of the selected records that were wrongly coded.
    - Small sample: small cost (manual coder workload) but potentially high outgoing error rate.
  - This method will be able to:
    - Asses the incoming error rate (IER) and outgoing error rate (OER);
    - Monitor model deterioration through time or determines if new data behaves differently than test/validation data.

Incoming Error Rate

- Estimated overall **incoming error rate** ($\widehat{IER}$):
  - The proportion of errors found during the QC process (for each coder and for the entire coding exercise).
  - The error rate is essentially the sample mean ($\bar{y}_s = \hat{p}$).
  - Each coder's incoming error rate takes into consideration the errors among the records reviewed in QC. The overall incoming error rate is a stratified SRS where each coder is a stratum:

$$\hat{p} = \widehat{IER} = \sum_{h=1}^{H} \frac{N_h}{N} \widehat{IER}_h = \sum_{h=1}^{H} \frac{N_h}{N} \left(\frac{\sum_{j=1}^{n_h} E_{h,j}}{n_h} \right)$$

Where $E_{h,j}$ is a 0/1 indicator that assigns a value of 1 to the *j*^th record coded by coder *h* if the code assigned by the coder does not match the final code assigned.

  - A 95% confidence interval can also be provided, calculated as follows:

$$\widehat{IER} \pm 1.96 * \sqrt{V(\widehat{IER})}$$

Outgoing Error Rate

- Estimated overall **outgoing error rate** ($\widehat{OER}$):
  - The outgoing error rate is estimated by determining an estimate of the number of errors included in the records that were **not verified** and dividing this by the total number of records:

$\widehat{OER}$ = (estimated # of errors among not reviewed in QC) / (total # records coded)

$$\widehat{OER} = \sum_{h=1}^{H} \frac{N_h}{N} \widehat{OER_h} = \sum_{h=1}^{H} \frac{N_h}{N} \left(\frac{\widehat{IER_h} * (N_h - \mathrm{n}_h)}{N_h} \right)$$

  - A 95% confidence interval can also be provided, calculated as follows:

$$\widehat{OER} \pm 1.96 * \sqrt{V(\widehat{OER})}$$

Example: Incoming and Outgoing Error Rates

- Example:
  - 100 records to be coded for a survey X
  - 20 records are verified by an auditor (QC Sampling Fraction : SF_h = 20%)
  - **What's the estimated incoming error rate?**
    - 5 errors are identified among the 20 verified records.
      - Estimated incoming error rate = $\widehat{IER}$ = 5/20 = 25%
  - **What's the estimated outgoing error rate?**
    - 80 records remain unverified.
    - From the estimated $\widehat{IER}$, we know that 25% of codes are erroneous.
    - Estimate 20 errors among the 80 remaining codes (25%).
      - Estimated outgoing error rate = $\widehat{OER} = \frac{\widehat{IER}*(N-n)}{N} = \frac{0.25*(100-20)}{100} = 0.2$ = 20%

Sampling Fractions

- Currently, Statistics Canada specifies sampling fractions using the traditional formula for estimating a proportion (*p*) for a Simple Random Sample (SRS) for a required precision (*e*) and a specified level of confidence (*z*):

$$\text{n} = \frac{z^2 p(1-p)}{e^2 + \frac{z^2 p(1-p)}{N}}$$

- Coder (*h*) sampling fractions are calculated using:
  - An estimate of the coder's incoming error rate ($\widehat{IER}_h$), e.g., an estimated incoming error rate from the previous cycle.
  - An estimated workload, $\widehat{N}_h$:

$$SF_h = \frac{n_h}{\widehat{N}_h}, \text{ where } n_h = \frac{z^2 \widehat{IER}_h (1-\widehat{IER}_h)}{e^2 \frac{z^2 \widehat{IER}_h (1-\widehat{IER}_h)}{\widehat{N}_h}}$$

- Typically, z and *e* values are specified to manage the overall workload for the Statistics Canada Coding Centre.
  - Often, z = 1.96 and e = 3% are used in production.

Simplex Optimization for QC Sampling Fractions

- Without AOQL, it is not possible to ensure an outgoing error rate.
- Under the SRS approach, the goal becomes **how to select a sampling fraction** for each coder in order to **minimize** the overall outgoing error rate, **given a manual coder budget.**
- Since 2022, the Labour Force Survey sampling fractions were specified for each coder using a **Simplex Algorithm** (or Simplex Method, developed by G.B. Dantzeg, 1947) used for linear programming.

Simplex Optimization for QC Sampling Fractions

- **Minimize** the overall outgoing error rate (objective function):

$$\widehat{OER} = \sum_{h=1}^{H} \frac{N_h}{N} \widehat{IER}_h (1 - SF_h) = \frac{N_1}{N} \widehat{IER}_1 (1 - SF_1) + \ldots + \frac{N_h}{N} \widehat{IER}_h (1 - SF_h)$$

    - Where N is the total number of records to be coded, N_h is the workload of coder h, $\widehat{IER}_h$ is an estimate of the coder error rate and SF_h is the sampling fraction for coder h.

- Subject to:
    - *Constraint type # 1*: Upper and lower bounds for coder sampling fractions

$$LB_1 \leq SF_1 \leq UB_1, \ldots, LB_h \leq SF_h \leq UB_h$$

    - Where naturally, $LB_1, \ldots, LB_h \geq 0$ and $UB_1, \ldots, UB_h \leq 1$.
    - However, these bounds are selected so that an accurate estimate for each coder's incoming error rate can be calculated.

Simplex Optimization for QC Sampling Fractions

- *Constraint type # 2*: manual coder workload budget

$$SF_1 * N_1 + \ \dots + \ SF_h * N_h \leq \boldsymbol{C}$$

  - Where *C* is calculated based on the workload budget, the estimated number of records to code and the expected rate of second verifications.

- Example to calculate C:
  - Targeted workload = estimated # of records to code + # verifications + expected # of second verifications = 25,000
  - Estimated # records to code = 18,000
  - Expected second verification rate = 40%
  - C = (25,000 – 18,000)/(1.4) = 5,000
  - Therefore, 18,000 coded records + 5,000 verifications + 2,000 second verifications = 25,000

- The Simplex Method can be implemented:
  - In SAS, using *proc optmodel*
  - In R, using the library *lpSolve*

Results

- The following table presents average values observed for each coding cycle observed to date in 2022.

Survey Program	Average # Records to Code	Average Overall Inspection Rate	Average Autocoding Rate	Incoming Error Rate		Outgoing Error Rate	
				Manual Coders	Autocoder	Manual Coders	Autocoder
Labour Force Survey (LFS)	20,000	25%	20%	10-20%	1%-5%	10%-20%	1%-5%
Jobs Vacancy and Wage Survey (JVWS)	100,000	20%	40%	10-20%	1%-5%	10%-20%	1%-5%
Building Permits Survey (BPER)	25,000	30%	20%	10-15%	15%-20	5%-15%	5%-15%

Conclusion

- Quality Control is an important step of a machine learning activity implementation that is often forgotten.
  - It ensures that we can measure the outgoing error rate produced by the model;
  - It could help with the ML model drift that will likely happen in 2+ years;
  - It corrects data from the sample that was wrongly coded by the ML process.
- Many QC methods exists:
  1) SRS QC
  2) AOQL
- Programs should at least produce QA statistics and scores:
  - Coding rates, precision/recall error rates (with test/validation data) and F1-Score.

Acknowledgements

We would like to thank the following contributors :

- Justin Evans (Statistics Canada, OID)
- Julie Portelance (Statistics Canada, OID)
- Anthony Yeung (Statistics Canada, SIMD)
- Scott Wile (Statistics Canada, NEAD)

Questions

Javier Oyarzun

Javier.Oyarzun@statcan.gc.ca

Laura Wile

Laura.Wile@statcan.gc.ca

Thank you!