Automated Coding using the IMF's Catalog of Time Series

This pilot study – conducted by the IMF Statistics Department[1] - aimed to automate the coding of time series collected from member countries into the IMF's Catalog of Time Series (CTS), an internal nomenclature of descriptors and mnemonics of economic and financial variables used in the policy framework of the IMF. Member countries participating in the IMF's Data Dissemination Standards Initiatives (SDDS Plus, SDDS, and e-GDDS) publish economic time series data on their National Summary Data Page (NSDP). IMF staff code these series according to the CTS. The coding process is manual and time consuming. The objective of this pilot study is to create an automated solution based on machine learning techniques that can assist this coding process. The coding will be done using the descriptors of indicators provided by the countries and some of the time series features.

1. Data source

The IMF Statistics Department assists member countries to implement the IMF's Data Dissemination Standards Initiative (SDDS Plus, SDDS, and e-GDDS). These standards require or recommend the dissemination of selected data through the country's NSDP using the SDMX format (currently only for e-GDDS and SDDS Plus countries). Data published on the NSDP need to be coded using the internal coding system to facilitate the ingestion of the data for and subsequent internal use at the IMF. Over the years, the IMF has accumulated thousands of economic indicators coded into the CTS hierarchy. The association between indicators and CTS codes has been done manually based on the indicator descriptor and with the help of subject matter experts.

For this study, data files were combined to create a labeled dataset with thousands of time series matched to the corresponding CTS code. Only series with English descriptors mapped to the CTS were included in the training and test data for this pilot study (some countries provide non-English descriptors).

- Number of series collected **~100,000**
- Percentage of series mapped to CTS code: **~ 57%** (potential labeled data)

Below is an example of a country data file submission via NSDP, using the ECOFIN Data Structure Definition:

- Time series attributes:

DATASTRUCTURE	IMF:ECOFIN_DSD(1.0)	Datastructure
DATASTRUCTURE_NAME	ECOFIN Data Structure Definition	Datastructure name

[1] November 2020. Report prepared by the data analytics team of the IMF Statistics Department (Ayoub Mharzi, Alberto Sanchez, Alessandra Sozzi, Pat Escalante, Yamil Vargas, and Marco Marini). The views expressed in this note are those of the authors and do not necessarily represent the views of the IMF, its Executive Board, or IMF management.

DATA_DOMAIN	NAG	Dataset
REF_AREA	AE	Country
COUNTERPART_AREA	_Z	Counterpart area
UNIT_MULT	6	Scale = Million
FREQ	A	Frequency = Annual
COMMENT		Observation status

- Time series descriptor and data values

Descriptor	INDICATOR	BASE_PER	2013	2014	2015	2016	2017
Nominal GDP by Activity	?	?	1432669.8984	1480521.3947	1315250.5834	1311248.3355	1405006.8341
Agriculture, forestry and fishing	?	?	9223.0676	9468.2351	9746.3480	10175.8220	10721.0735

Catalog of Time Series (CTS)

The CTS provides a standard framework for the structure, nomenclature, and coding of economic indicators and times series used in the IMF. It consists of a list of economic concepts and codes as well as a set of coding rules. Standardized codes help improve data management practices and facilitate Fund-wide data sharing. The CTS is the authoritative source for economic concept codes used within the Fund. It is the main reference for IMF flagship databases such as the World Economic Outlook (WEO), the International Financial Statistics (IFS), and other IMF regional and functional department databases.

Currently, there are 28,919 CTS codes. A few examples are provided below.

- Descriptors and codes:

Code	Full Descriptor	Methodology Reference	Sector - Name	Topic - Name
NGDPVA	National Accounts, Activity, Memorandum Items, Gross Value Added, Nominal		National Accounts	Activity
NGDPVAGA	National Accounts, Activity, Memorandum Items, Gross Value Added, of which Government Activities, Nominal		National Accounts	Activity
A_CPC21_0	Economic Activity, Production, By Central Product Classification (CPC) Version 2.1, Agriculture, forestry and fishery products	FAO SEEA AFF; CPC Version 2.1	Economic Activity	Production
ACO_CPC21_0	Economic Activity, Consumption, By Central Product Classification (CPC) Version 2.1, Agriculture, forestry and fishery products	CPC Version 2.1	Economic Activity	Production

- Attribute and codes:

Code	Name
_SA	Seasonally adjusted
_XDC	Domestic Currency

2. Machine learning solutions

We experimented with different machine learning techniques in this pilot study:

- **Feature extraction**: we used two techniques to transform the descriptors of country indicators into numerical features usable for machine learning.
    - **TF-IDF**: a word weighting scheme where descriptors are converted into fixed-length vectors based on word occurrences. Common words across all descriptors are penalized by assigning them lower weights while more importance is given to rare words.
    - **Word2Vec**: family of models that try to represent each word in a large text as a numeric vector in an N-dimensional space. We used the FastText library (also used by Statistics Canada) to implement one of these models: Skip-gram. Skip-gram takes every word in a large corpora (we will call it the focus word) and also takes one-by-one the words that surround it within a defined 'window' to then feed a neural network that after training will predict the probability for each word to actually appear in the window around the focus word. Intuitively, the model will generate similar vectors for words that share the same context words. We then average these word vectors for each of the words appearing in a descriptor sentence to obtain a single descriptor vector.
- **Supervised Machine Learning Models**: we used two methods to learn from historical manually-coded descriptors of country indicators (here converted into numerical vectors) and assign the right code to new unseen descriptors.
    - **Multinomial Logistic Regression**: a parametric method that estimates parameters from the historical data and assigns a probability to every possible code the descriptor can take. The code that gets the highest probability is then selected.
    - **Nearest Neighbor**: a non-parametric method that assigns a code to a new descriptor by searching among the historical data the closest descriptor. Closeness in this case is measured by cosine similarity that measures the cosine of the angle between vectors of descriptors.
- **Time series clustering**: the Dynamic Time Warping (DTW) algorithm was tested to measure similarity between the actual time series of the indicators. We used this approach to cluster features of time series that can provide additional insights to the coding task. For example, we could identify series characteristics such as seasonally adjusted, growth vs. volume, etc. to narrow down the predicted code for series where this information is missing from the descriptor.

We built 4 different types of machine learning solutions from combinations of feature extraction techniques and the chosen machine learning models:

- Nearest Neighbor with TF-IDF
- Logistic Regression with TF-IDF
- Nearest Neighbor with Word2Vec

3. Key steps, facilitators and challenges

Data preparation

Countries submit time series data to the IMF in a variety of ways and, at times, in a variety of formats too. We use a semi-automated approach to convert these data into a standard format. Once standardized, we transform the data into an input-ready format for learning models. For example, we compute word embeddings or tf-idf matrices.

- Facilitators: The team's experience and knowledge of the different country file formats and exceptions. The data science skills across the team members.

- Challenges: The different country file formats make data processing prone to errors which require manual intervention and it is time consuming. Some country files were left out as they are only available in national language.

Identification of ML techniques

The most informative data feature is the indicator descriptor. We approached this exercise as a natural language programming (NLP) problem. Once we transformed text into numeric vectors, we used simple techniques to approach how the model will learn to assign codes to descriptors. The logistic regression and nearest neighbors models were used to predict the closest code of a given descriptor from the training data.

In addition, country files contain time series data so we used time series clustering techniques to compute a distance measure and apply nearest neighbors. The idea was to help the text-based methods in cases where the descriptor does not provide enough information to make a good prediction. For example, classify input series as seasonally adjusted.

- Facilitators: The nature of the data, once transformed, allows for well documented techniques which are straightforward to apply.

- Challenges: The data is heavily imbalanced across data domains (variety of descriptors), which results in poor predictions for domains with smaller sample sizes. The time series clustering approach is experimental and not as well documented and tested as the other methods.

Training/Testing

We split the standardized data in 90 percent for training and 10 percent for testing. To speed up the process, we did not use cross-validation to train our models, we tested a few parameters and checked for consistency of results across different samples.

- Facilitators: Because we did not do proper cross-validation, this simplified this part. The imbalanced nature of the data helped get consistent in-sample results across different training samples.

- Challenges: We did not invest enough time to do proper cross-validation although we are confident that different parameters for our models would not have made a difference.

Assessment of results

We used overall accuracy to assess the results. Specifically, we looked at the predicted top 10 codes from each of the models to come up with a prediction model that combines the strengths of the different base models. For production stage, we envision a model that, for each new input will return a list of potential codes ranked by probability. In some cases, a human will have to eventually decide which code to take from the list.

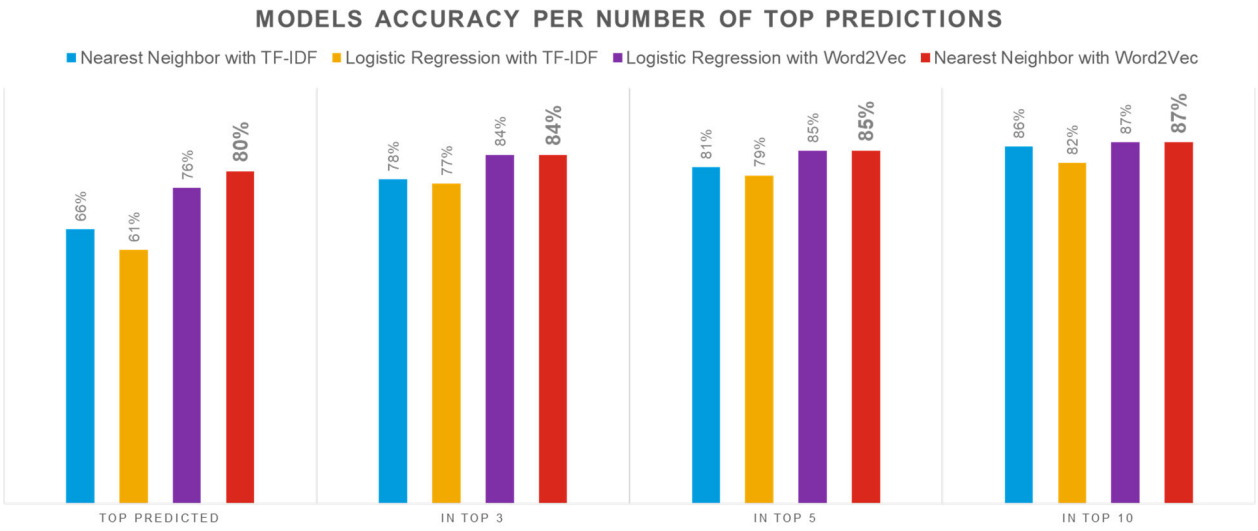
- Facilitators: Applying different transformations and models helps extracting features from the data and a voting system provides more consistent results.

- Challenges: The model by committee approach will not fix the inherent flaws in the data, namely the lack of representatives in the sample for certain descriptors. We did not use standard measures of model assessment such as precision, recall and F1.

4. Results

As shown in Figure 1, the Word2Vec extraction method provided the most accurate predictions, with the Nearest Neighbor being the better model compared to the Logistic regression.

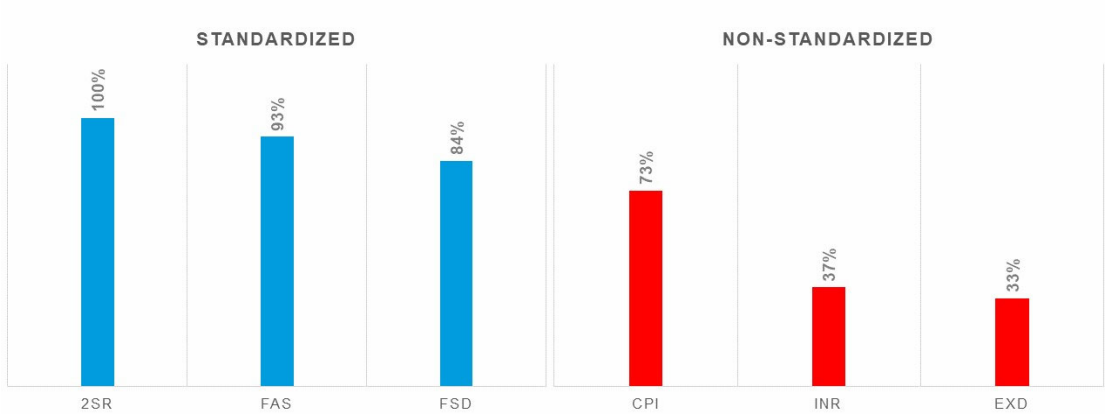
Figure 1: Models accuracy per number of top predictions



Other considerations on the key results of our study follow in order:

- Results by domain showed different levels of accuracy rates.
    - Predictions for domains where standardized report forms are used were far more accurate than for domains using non-standardized report forms (see Figure 2). Standardized report forms contain well-structured descriptors for the series, reducing variability between countries.
    - The limited number of examples for indicators from non-standardized report forms also impacted the prediction accuracy for selected domains.
- Time series descriptors contain valuable information to predict codes. A full descriptor of the indicators is essential to improve accuracy.
- An ensemble model can help improve the quality of the results. Accuracy results by different groups of indicators (e.g. by domain) showed instances where models performing worse overall outperformed Nearest Neighbor and Word2Vec.

Figure 2: Accuracy for selected domains using standardized vs. non-standardized report forms. Results from Nearest Neighbor and Word2Vec.



5. Collaboration

Experience from other organizations

The experience from the U.S. Bureau of Labor Statistics (BLS) in the project was very helpful for us to narrow down the possible techniques to approach this problem.

We were also inspired by the work of Statistics Canada and their use of the *fasttext* library to implement text to vector techniques. Both collaborations were critical in reducing research time and allowed us to spend more time on data processing.

Teamwork

The pilot study benefited from collaboration of various team members with country data knowledge, data science skills, and coding expertise.

6. Lessons learned

From this pilot study, we expect benefits in the following three areas:

- Faster coding

  Machine learning can speed up the process of assigning CTS codes to new indicator series. These may result in savings of several days based on the estimated completion time of current manual coding.

- Better accuracy

  The use of an automated system will help increase the accuracy and consistency of the coding and reduce the subjectivity inherent in human-based coding.

- Reduced costs

An automated mapping of the CTS codes will free up valuable time for our coders, who can spend more time on higher value-added tasks.

There are two main lessons learned from the IMF in this pilot study:

- Human intervention is required to ensure that the code produced by machine learning models is correct. We envisage the final product to be a production tool that automates coding when the model has high confidence and aides the coders with likely candidates when confidence is not as high. Human coded series can then be fed back into the training data to improve performances of the model over time.

- Having a team which combines different skill sets and expertise is key to be able to incorporate subject matter knowledge and user needs while providing the necessary data science skills required to understand, process and analyse the data and develop tools to interact with the data.

7. Deliverables

We envisage the final product of the pilot study to be an application tool for internal users working on classifying macroeconomic time series based on CTS (non-technical users). It is meant to be used as a system to ingest country files, classify them on the fly and provide an interface to assess (accept/reject) the codes predicted by the application tool. Internal users are also presented with options that could contain the correct choice, instead of a single CTS code. The application tool will be available for internal use.

The report and presentation of this pilot will be made available in the group's Wiki. The algorithm and code developed for the pilot study – when finalized and tested – will also be available. In the meantime, we can share it with other organizations upon request.

8. Next steps

We will expand our dataset with indicators from non-standardized report forms to increase representativity of our sample.

We trained our models using default values for their parameters. We will train our models using proper cross-validation to select optimal parameters.

For our study, the Word2Vec feature extraction with Nearest Neighbor provided the best results. However, in some cases other combinations of feature extraction methods and models provide better results. We will test the combination of different models to improve the quality of our predictions.

Our ultimate goal in this project is to develop a tool to assist in the CTS mapping work. A key step to move from the testing phase to production will be to establish quality control thresholds for the prediction of codes.

Embedding machine learning solutions in internal coding work will lead to efficiency gains and free up resources to perform higher value added tasks. This experience can also demonstrate the utility of machine learning for repetitive and predictable work processes in the IMF, and encourage other teams to test machine learning tools for automation.