Hierarchical Multi-Level Spending Classification

Riccardo Ricci | SIADS 699 Capstone Project | University of Michigan | github.com/Rick0701/Hierarchical_Multi_Level_Spending_Classification

UNIVERSITY OF MICHIGAN

UMSI

Background

- Spending classification can be tedious, expensive and sometimes even difficult to apply for workers in procurement areas;
- Making mistakes can cause inefficiencies in resource allocation impacting business operations. Implementing this task with an automatic solution such as ML can be extremely beneficial for the client.

Problem Statement

- **Goal**: predict Purchase Orders (PO) spending categories at four hierarchical levels;
- **Challenges**:
  - many labels to assign at different hierarchical levels with greater complexity going down the tree;
  - Short PO descriptions (avg 3 tokens);
  - Ambiguous labels in training data due to wrong users imputation in source system. Test set certified by users but unbalanced (see to the right) → ground truth

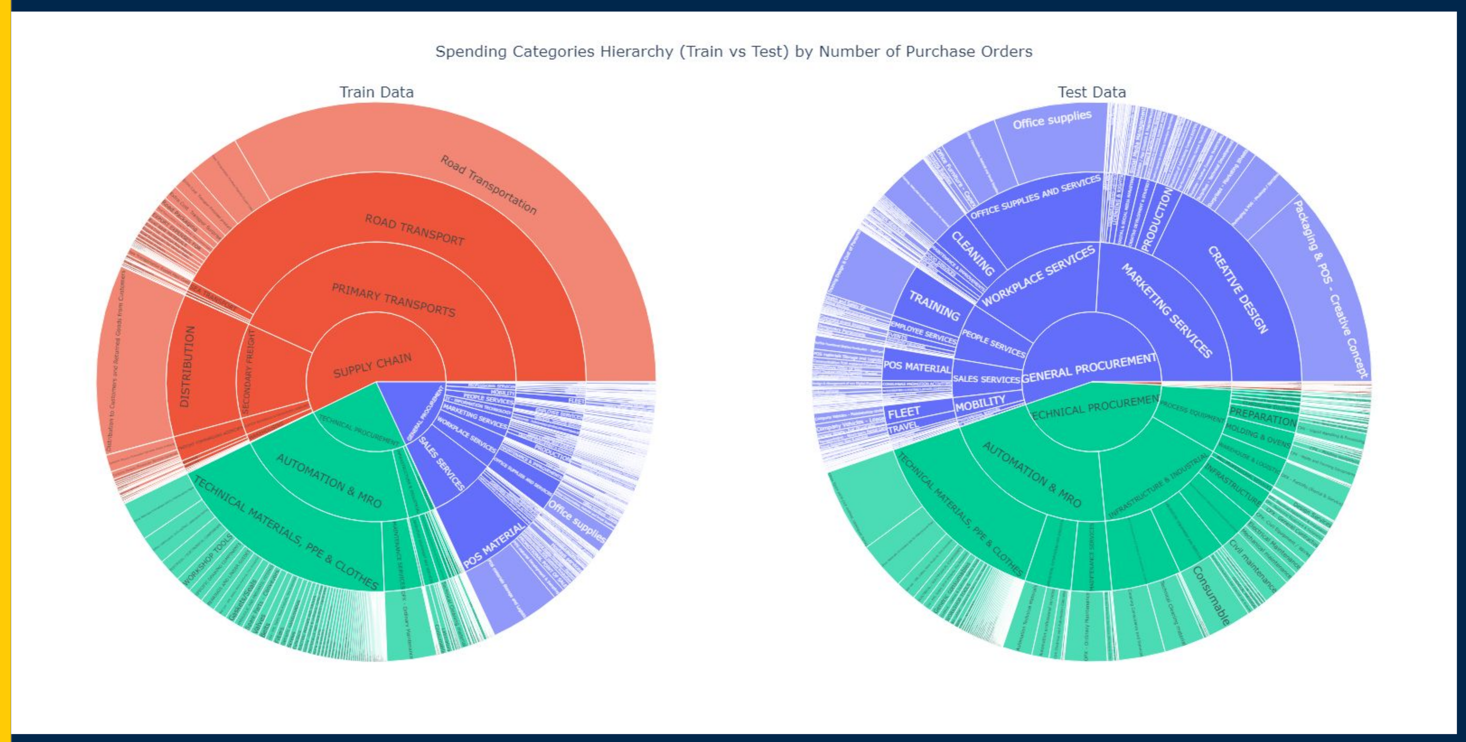
Level	Number of Categories	Accuracy Baseline
Level1	3	0.95
Level2	20	0.80
Level3	104	0.70
Level4	573	0.65

Methodology

- Multi-step machine learning models that integrate previous predictions made at higher levels;
- Classifiers (*scikit* learn library):
  - Decision Tree
  - Random Forest
  - Gradient Boosting
- **Feature Engineering** (see below)
  - NLP, One-hot encoding with minimum frequency.

Feature	*Description*	*Data Manipulations*
Purchase Orders descriptions (see Figure 2)	Description of a PO, mostly short sentences and sometimes only composed by numbers or dates.	➢ Regex Syntax suitable for the data; ➢ Removal of stopwords, applied tokenization (using the custom regex); ➢ Stemming and vocabulary; ➢ Minimum frequency: limit the number of features, I removed tokens that occur less than 50 times; ➢ Text Representation (different approaches): ○ **TF-IDF**, passing the list of tokens (either using words *lemma* or ***stemma***) and the vocabulary as arguments; ○ Latent Semantic Indexing (output to TF-IDF) to further reduce the number of features; ○ Embeddings using Word2Vec to embed meaning.
Suppliers; Purchasing Organization, Purchasing Group, Company	➢ Suppliers: companies delivering a commodity or product; ➢ Purchasing organization, or group or company asking for that product or commodity.	➢ One-hot encoding; ➢ Minimum frequency is applied to remove uninformative features. This helps to make the model less complexing reducing overfitting: ○ Suppliers: at least 10 PO; ○ Purchasing Organization: 10 PO; ○ Purchasing Group: 10 PO; ○ Company: 20 PO.

Results

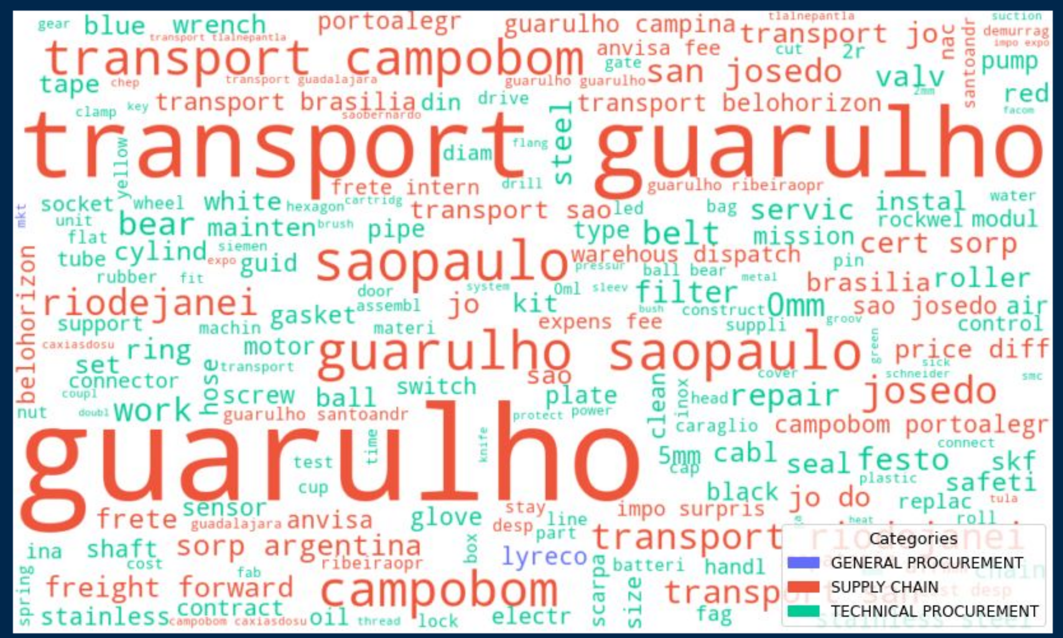


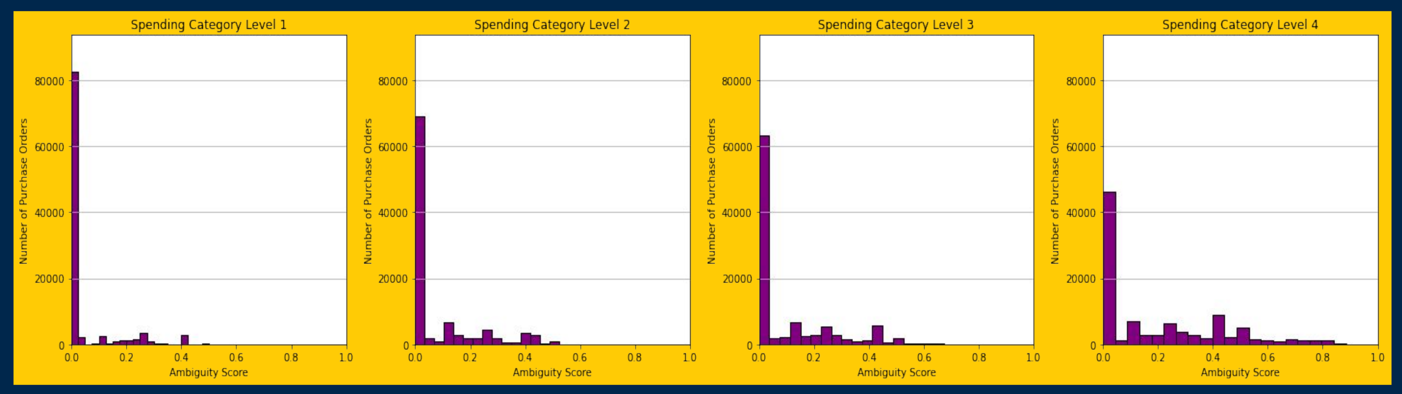
Train vs Test sets

- *Supply Chain* (SC) category underrepresented in the test set. *General Procurement* and *Technical Procurement* proportions swapped;
- **Shift in category distributions** suggests that a model trained on the original data may face ***overfitting*** when applied to unseen data unless **covariate shift adaptation techniques** are employed.

NLP on PO Description

- *Transport*, and its related bigram *transport guarulho* are the most used words and representative of the SC category signaling a discriminative power for this category;
- **SVD** for latent semantic analysis helps reducing the feature space and lowers ***overfitting*** risk but results in dense vectors, increasing **computational cost**;
- Pre-trained Word2Vec (Google News) does not improve performance, likely because PO descriptions are highly domain-specific;
- TF-IDF with lemmatization performs similarly to **TF-IDF with stemming**, but stemming is preferred due to lower computational requirements.





Ambiguity

As classification depth increases (from L1 to L4), the ambiguity score distribution shifts toward greater values → higher ambiguity and reduced discriminatory power.

	Level 1		Level 2		Level 3		Level 4	
	Macro-Avg	Weighted - Avg	Macro-Avg	Weighted - Avg	Macro-Avg	Weighted - Avg	Macro-Avg	Weighted - Avg
Decision Tree	0.87	0.97	0.65	0.77	0.67	0.71	0.53	0.70
Random Forest n_est= 10	**0.88**	**0.97**	**0.70**	**0.79**	**0.58**	**0.74**	0.55	0.72
Gradient Boosting n_est= 500	0.72	0.94	0.56	0.63	n.a. (1)	n.a (1)	n.a. (1)	n.a. (1)

(1) The model was not able to conclude in reasonable time due high computational requirements

Discussion

- **Well-performed NLP is necessary** for satisfactory performance of downstream tasks like classification:
  - Sometimes custom regex is necessary to capture domain-specific syntax;
  - SVD can reduce the feature space but at the expense of computational requirements (dense vectors).
- Old-fashioned classifiers like **Random Forest** can still provide **solid results** and more importantly provide **demonstrable results** to a **non-technical audience;**
- Develop a new easy-to-understand metric: **Ambiguity Score** which can **communicate to non-technical audience** why a model is not performing well for some categories;
- Develop a **framework for multi-level ML** i.e. **integrating prediction** of categories at leve *i-1* to predict categories at level *i*;
- Poor balance between training and test data and ambiguous labels of training data → the most important asset in any machine learning project is the data itself - its quality, representativeness, and balance largely determine the success or failure of the model. **No algorithmic sophistication can compensate for poor-quality data**;
- Model predictions risk **amplifying existing labeling errors and biases** due to incomplete and subjective training data, highlighting the **need for careful human oversight.**

Future Work

- **Translation of Untranslated Tokens** (e.g. *transport guarulho*);
- **Train domain-specific Word2Vec models** on the internal corpus of PO descriptions to capture idiosyncrasies;
- **Separate models for different categories** (only if labeled data is correct in the training data).

QR Code

Digital Access

Full project report