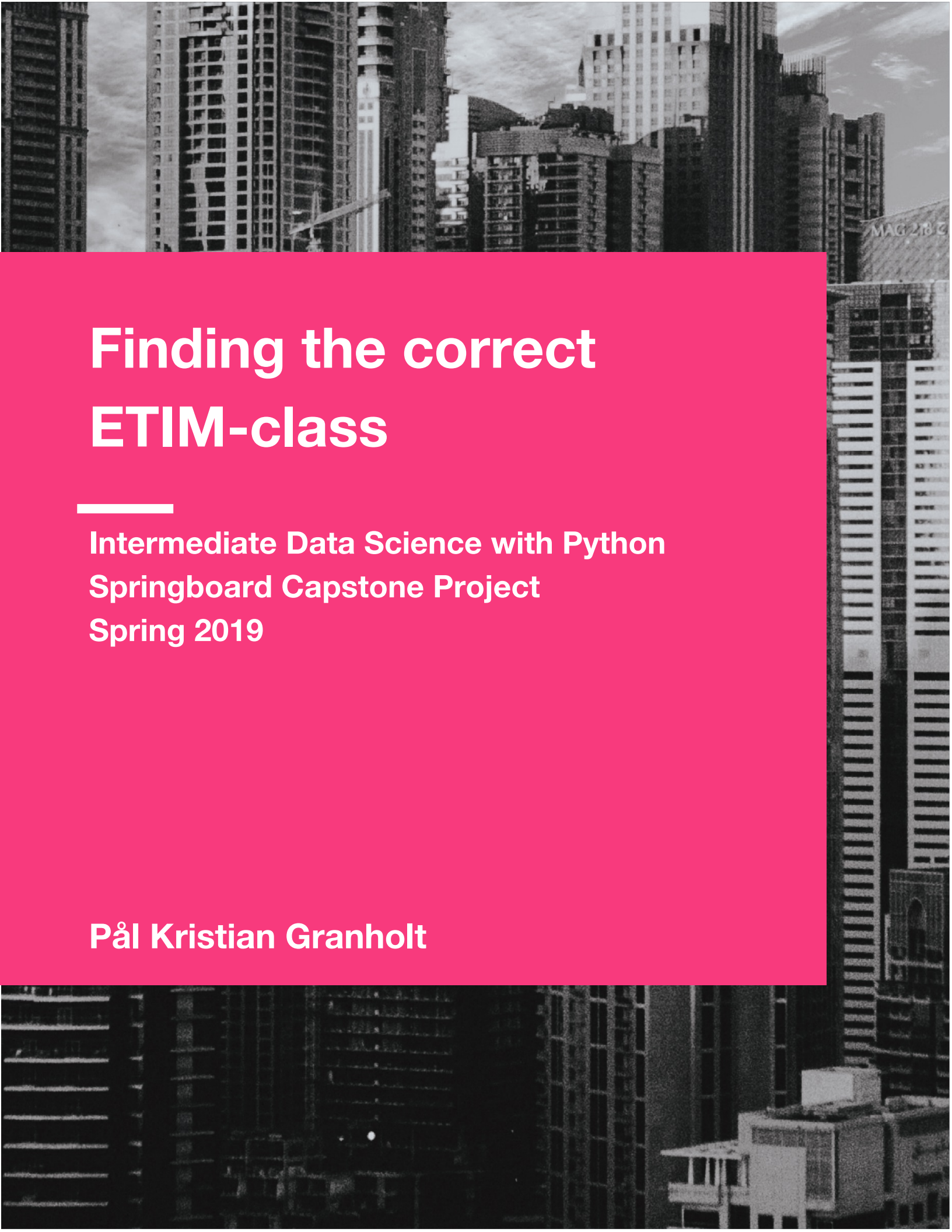

Finding the correct ETIM-class

Intermediate Data Science with Python
Springboard Capstone Project
Spring 2019

Pål Kristian Granholt

Table of Contents

Introduction

The Client

The Norwegian Electrical Trade Organization (EFO) works for the suppliers and wholesalers of electrotechnical equipment and solutions. This spans products from a single lightbulb to the largest infrastructure projects. With 500 member companies, their members' portfolios cover almost every type of electrotechnical product.

EFO has a product database that suppliers, wholesalers and installers use to communicate about products and their properties. There are about 300,000 products registered with EFO, most of which are active and currently accessible. As part of their mission to be the connector between the different roles in the industry, they have carefully selected the properties that products that are registered with them need to have. They evaluate the products that are registered with them to make sure they have the necessary data.

The Assignment

There are many standards that electrotechnical products need to be classified by. One of the most important ones is the ETIM (European Technical Information Model) classification model. There are at the time of writing 4725 ETIM classes, where 3041 of these are the most relevant for EFO's members. It is up to the member companies who own the products to correctly identify which ETIM class each product belongs to. There are tools to look up the different classes, but as a manual process, it can be time-consuming. EFO would like to assist in this task, by creating a machine learning model which can suggest which ETIM class each product belongs in.

In the product database, there can be several hundreds of features per product. Some examples of features are physical dimensions, logistical data, certifications and classifications. When one registers a new product, it's usually a good idea to start with the most critical features. The one prioritized in this project will be the ETIM class. An ETIM class describes on a general level what the product is. One example is the ETIM class for dimmers. There are many different types of dimmers, with many different features, made by many different manufacturers. But they can all be classified as dimmers.

Summary of results

The final product of this capstone project is a model that performs well both in regard to metrics for classification and testing from EFO domain experts. The accuracy of the model on the test data is 84%, but for a more comprehensive picture of the model performance, please refer to the Classification report section. The biggest hurdle for further improvement of the model is the large number of ETIM-classes with few products in the training data.

Approach

Data acquisition and wrangling

The data is taken from the EFO product database, and through an SQL-query only active products (about 238,000) were selected. Based on the domain knowledge from the experts at EFO, there are two factors that go in to the selection of features. The first is that the features are thought to be the most relevant for predicting the ETIM class. The second is that it should be features that are registered early and easily. In other words, if one already has classified a product in other classification models, adding the ETIM class is considered trivial. But this is not altogether a reasonable expectation, and I chose to only use a couple of features for the project:

Feature 1 - Electrical number group

All products in EFO's product database have a seven-digit unique identifier. The first two digits are what is called the Electrical Number Group (ENG), which is a very rough grouping of the products. For instance, if the first two digits are "10", the product is either a cable of some sort, or a product related to cables. The seven-digit identifier is by definition the first piece of information that is stored about any product in the EFO product database, so it is both early and easily registered.

The ENG is widely used in the industry, and if a product has been misclassified, it will most likely not be able to sell through wholesalers for instance, who are strict about having the correct ENG. For this project, it means I can assume the ENGs are correct without any further testing.

Feature 2 - Technical description

All products in EFO's product database have a text field with a description of the product and its features. This field goes years and years back and is a mandatory field for all products. It is therefore well known, and easily completed. Many member companies use standard text from their product catalogs, so for many, it is a copy-paste job. For others, it is simply a matter of describing the most pertinent features of their product, which all product managers can do with ease. The technical description is one of the first fields that are entered for all new products in the database.

There are few demands or restrictions on the technical description, so it's expected that it contains texts which differs greatly. The EFO domain experts also say that for similar products, the same text is sometimes used. The text itself can be a little technical, and usually includes the basic facts about the product. Here is an example (in Norwegian):

VDF/EMC frekvensomformerkabel med symmetrisk jordleder. Dobbel skjerming bestående av folie og flettet fortinnet kobberskjerm. Laget for å gi lavest mulig koblingsimpedans. Kan benyttes utendørs. For spenning 0,6/1Kv med en testspenning på 4000V

This is the description of a specific type of cable, which is explicitly mentioned at the end of the second word "-kabel". Additionally, there are many features described.

The technical descriptions will be read and processed to extract the words, numbers or abbreviations that help predict the correct ETIM class.

The first step in the wrangling was to create the Electrical Number Group (ENG) from the unique identifier (ProductNumber). The dataset now has one index and three columns.

ProductNumber	ENG	ETIM_class	Technical_description
1000000	10	EC003251	VDF/EMC frekvensomformerkabel med symmetrisk j...
1000001	10	EC003251	VDF/EMC frekvensomformerkabel med symmetrisk j...
1000003	10	EC003251	VDF/EMC frekvensomformerkabel med symmetrisk j...
1000004	10	EC003251	VDF/EMC frekvensomformerkabel med symmetrisk j...
1000005	10	EC003251	VDF/EMC frekvensomformerkabel med symmetrisk j...

Figure 1 - Table of the first five rows in the dataset right after import

All products that have null values are removed for this project. This is primarily missing ETIM-classes. When all missing values are removed, about 202,000 products are left to work on further.

Data storytelling and inferential statistics

Instead of just looking at the top five rows, let's look at some descriptive statistics.

	ENG	ETIM_class	Technical_description
count	202216	202216	202216
unique	41	1814	109429
top	43	EC000042	Spasial serien består av stål Bokser typeS44,S...
freq	20761	5630	1772

Figure 2 - Table of descriptive statistics after wrangling

The counts confirm that there are no missing values in the dataset. There are 41 different ENGs, and 1814 different ETIM-classes. The difference here is expected, as ENGs are a rough grouping, while the ETIM-classes are a finer, more detailed division of product groups. There are about 109,000 different technical descriptions, which confirms that there are some technical descriptions that are re-used for different products.

The most frequent ENG is 43 (which consist of controls, engine protectors and power switches), and it is used almost 21,000 times, which is almost 10% of the dataset. This skewedness will be examined further a little later. The most frequent ETIM-class is EC000042 (which is miniature circuit breakers) and is used about 5,600 times. Perhaps the ETIM-classes are skewed less than the ENGs? The most used technical description is used 1,772 times, which is a little incredulous, could that many products really be described by the same text?

ETIM-classes and ENGs

The ETIM-class is what the final model is going to predict, it's the dependent variable. Let's look a little closer at this variable in the dataset. To get a sense of the distributions of the frequencies, here are two histograms, showing ETIM-classes and ENGs respectively.

Figure 3 - Histogram of frequency of the ETIM-classes

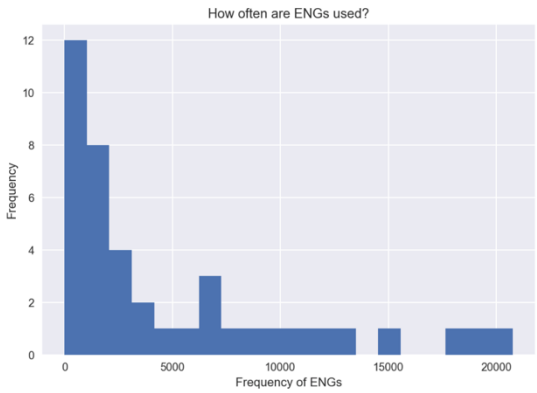

Figure 4 - Histogram of frequency of the ENGs

		freq
ENG	**ETIM_class**	
10	**EC003250**	5146
	EC003248	4819
	EC003251	2845
	EC003249	1779
	EC000034	1195
	EC000339	1144
	EC000405	639
	EC000839	288
	EC000019	261
	EC001476	207
	EC002063	136
	EC002065	91
	EC000247	67
	EC001139	66
	EC000838	65
	EC000065	62
	EC000340	58
	EC002904	31
	EC001474	18
	EC000441	11

Figure 5 - Table of 20 most frequent ETIM-classes within ENG 10

There's clear skewing of both the ETIM-classes and the ENGs. How does this look when both variables are taken into account? Let's look at the top 20 most frequent ETIM-classes within ENG 10.

The skewedness is still visible, and though it's not shown here, the tail of this table is quite long, with a lot of ETIM-classes occurring once. The good news is that there are classes with plenty of products. The bad news is that it will be difficult to build a model that is able to take advantage of the classes that occur as seldom as once.

The preferred method of fixing this would be to obtain more data for those seldom used classes. In practice for this particular problem, that will not be possible in the short term. Therefore, it will need to be dealt with in a different way. Possible options are undersampling (though not when the lowest frequency is one), oversampling or subsampling less skewed classes. These options are discussed later in the report.

For ENG 10, the pattern is a clear skewedness, but how is the distribution of frequencies for the other ENGs? As there are 41 of them in use, it would be daunting to view them all in a table form. As an alternative, figure 6 is a line graph that displays the frequency of the ETIM-classes, sorted by ENG and frequency within each ENG. The individual ENGs are hard to discern, but the patterns are easily visible; namely that there is skewing within all the ENGs, and almost all of them have rather long tails, where the frequency for the ETIM-classes in question

are one. This confirms what was visible from figure 5, skewdness is the rule rather than the exception in this dataset.

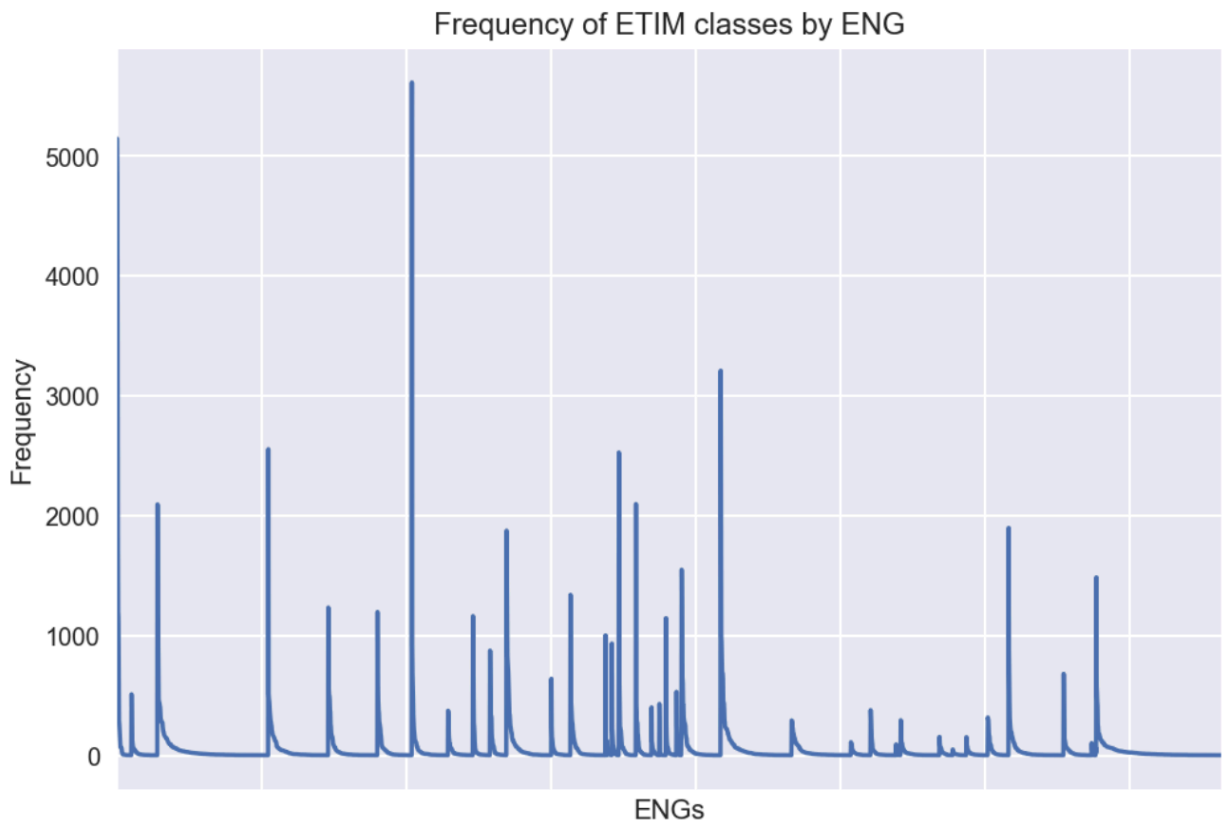

Figure 6 - Frequency of ETIM-classes by ENG, sorted by ENGs and frequency of ETIM-classes within each ENG

Since the rarer ETIM-classes will create some complications in the modelling, I thought it would be interesting to see if there are any natural cut-off points for the frequency of the ETIM-classes. To evaluate this, I created a graph that displays the fraction of data left if one adjusts the lowest frequency cut-off. The graph should be read like this: Assuming one selects a frequency on the x-axis as the lowest frequency of ETIM-classes that is acceptable (and thusly kept in the dataset), the corresponding value on the y-axis is the fraction of data that is still in the dataset. For example, if the cut-off is 1000, the dataset loses about 70% of all the data it originally had.

The smooth, sharp decline of the graph tells us that there is no clear cut-off point in the data. It also confirms what was visible in figure 6, that even a relatively low cut-off point will remove a large fraction of the dataset. If the cut-off was at a frequency of 50, about 1/6 of the dataset would be discarded.

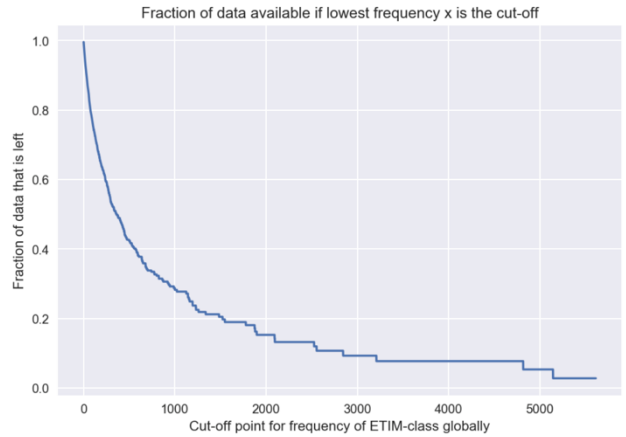

Figure 7 - Fraction of data available if the value on the x-axis is the frequency cut-off point, where all ETIM-class frequencies below this point would be discarded

Figure 8 - Same graph as figure 7, but with a narrower range on the x-axis to see if there are any differences that are visible at the lower end of the x-axis

Technical descriptions

To get a sense of the technical descriptions, let's also plot the frequency distribution. From the descriptive statistics, we know that there are about half as many unique technical descriptions as there are actual technical descriptions. So how is the distribution?

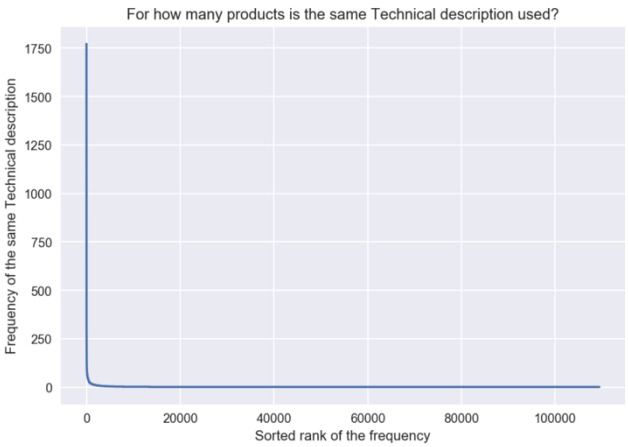

Figure 9 - Frequency distribution of the technical descriptions

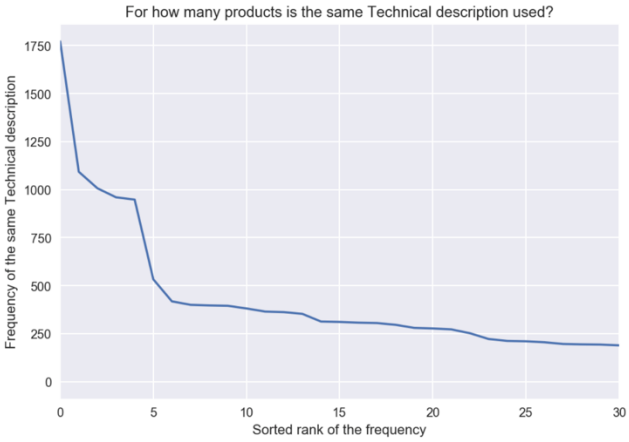

Figure 10 - Same as figure 9, but with a narrower range on the x-axis to see if there are any differences that are visible at the lower end of the x-axis

A clear pattern emerges; there are a few technical descriptions that are heavily re-used, while most of them are used only once. When the thirty most used technical descriptions are zoomed in on, there seems to be some variation on the first five, and after that there is a gradual decline.

A closer examination of the most used technical descriptions reveals (perhaps not surprisingly) that they are used by the same company for a range of products that are similar. When going through the ten most frequent technical descriptions, I noticed that not all of them were actually descriptions, two of them were in fact filler text. At this point, I ran various queries to find other examples of filler text. To find these filler texts, I searched for technical descriptions that were used in several ENGs, which indicates that products with identical text are not in fact related. I also checked for very short technical descriptions. In the end, I

found three technical descriptions that were just filler text. It was important to remove these, as they were used in several ETIM-classes and ENGs but did not actually contain any information. The removal of these technical descriptions reduced the dataset by almost 1,000 products. The remaining dataset is now in the excess of 201,000 products.

Inferential statistics

I have shown that there is skewedness in the ETIM-classes, the ENGs and the technical descriptions. I wanted to investigate if this affected the length of the list of unique words in the different ETIM-classes. To simplify, I extracted the five most used ETIM-classes:

ENG	ETIM_class	freq
16	**EC000042**	5615
10	**EC003250**	5146
	EC003248	4819
43	**EC000228**	3210
10	**EC003251**	2845

Figure 11 - Five most frequent ETIM-classes

I then found the dictionaries for each of these five classes separately and counted the lengths of them. These are plotted in figure 12. In addition, I did a simple regression with a fitted line. The dictionary length was the dependent variable, and the frequency of the ETIM-class was the independent variable. My thought was; if one more product is added with this ETIM-class, how would I expect the dictionary length to vary?

The answer for the five most frequent ETIM-classes seems to be that each additional product in the class, adds just over one additional unique word to the dictionary.

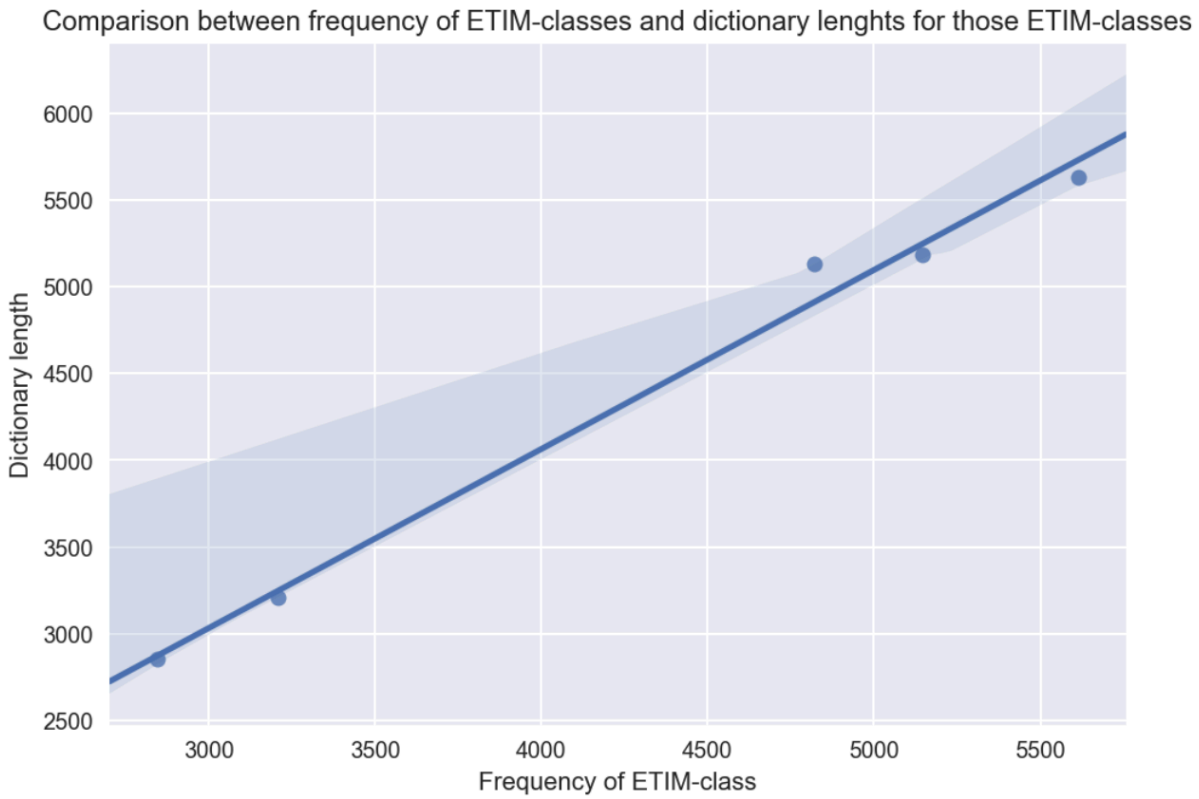

Figure 12 - The five most frequent ETIM-classes, plotted against their dictionary lengths (unique word count)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.999
Method:                 Least Squares   F-statistic:                     6210.
Date:                Thu, 02 May 2019   Prob (F-statistic):           1.55e-07
Time:                        14:55:03   Log-Likelihood:                -30.834
No. Observations:                   5   AIC:                             63.67
Df Residuals:                       4   BIC:                             63.28
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             1.0181      0.013     78.804      0.000       0.982       1.054
==============================================================================
Omnibus:                        nan   Durbin-Watson:                   2.398
Prob(Omnibus):                  nan   Jarque-Bera (JB):                1.750
Skew:                         1.446   Prob(JB):                        0.417
Kurtosis:                     3.184   Cond. No.                         1.00
==============================================================================
```

Figure 13 - Regression results

The p-value is extremely low, which indicates that this result did not occur by chance. This is also what one would expect by looking at figure 12.

Baseline model

The model is using bag-of-words to evaluate the technical descriptions. This means each technical description is converted to a vector of numbers, where each number indicates the number of times each word occurs in the text. This is done for all technical descriptions, which means that the vector includes all possible words in all technical descriptions.

I say words, but in reality, there are many nuances to the way various phrases are handled. Several operations are done to ensure maximum likeness for phrases that may be spelled differently. For instance, the phrase “VDF/EMC” is converted to the words “vdf” and “emc”. The slash sign is removed, and the two words are in lower caps. The phrase “0,6/1Kv” is converted to simply “1kv”. There are many other examples, but this illustrates that there could be a lot of tweaking at this point to ensure that specific phrases are better retained. For this project, I have opted to use default options for the vectorizer that creates the long vector for all the words. The ENGs have been converted to dummy-variables.

Bag-of-words is a simple way of extracting meaning from a sentence. It does not evaluate a sentence as a whole, but rather as occurrences of terms. It won’t be able to distinguish meaning between phrases like “good” and “not good”, because it only counts words. In the extended model, I tried to increase the words that were seen together to retain more meaning from the sentences.

After the bag-of-words transformation, I have used multinomial Naive Bayes to predict the correct ETIM-class based on the features from the technical descriptions and the ENGs.

I have used a 70-30 split between the training and the testing data. This means a random 30% of the dataset was set aside before any training was done, and all training was done on the remaining 70%. After the model was finished, the model was evaluated on the test data (30%).

The baseline model is without hyperparameter tuning, everything is set to default. It provides a benchmark to beat for the various adjustments for the extended model. It performed a little better on the training data than the test data. If a model performs much better on the training data than the test data, it would be overfitting and that can be problematic. That would mean that the model has learned random patterns from the training data that are not applicable to data outside the training set. In this case, the difference was small, so it's not problematic.

Extended model

For the extended model, hyperparameter tuning was done. The final hyperparameters included:

- Laplace-smoothing
  For the multinomial Naïve Bayes classifier, a very small value was used for alpha.
- Stop words
  Stop words are all the little words that are used often but are expected to carry little meaning in prediction purposes. In English, a few examples are "a", "the" and "or".
- ENGs
  The removal of the ENGs made no difference to the model.
- Ngram-range
  The ngram-range was set from one to two. This means the vectorizer not only includes single terms, but also two terms in succession. "Powerful flashlight" would be vectorized to "powerful", "flashlight" and "powerful flashlight". This allows bag-of-words to extract a little more meaning of the context the terms are in.
- Minimum term
  Ignores terms that happens only once across all technical descriptions. Has no effect on the classification report, but greatly reduces the runtime of the model when the ngram-range is increased.

The addition of the ngram-range from just one (regular bag-of-words) to one to two increased the terms significantly, but the inclusion of a minimum term reduced it a little. The Laplace-smoothing, stop words and the increased ngram-range all made contributions to improve the model. Hyperparameter-tuning of these values ended up with the best results I have been able to get.

Analysis of results

Classification report

To evaluate the model, I'm looking at the classification report. This reports three numbers; the precision, the recall and the f1-score. The precision is the predicted products that are correctly identified. This word "accuracy" is often used in its stead. The precision is the number of true positives divided by all predicted products. The recall tells us how many of the correct class the model predicted. Finally, the f1-score evaluates both the precision and the recall, to give one number that measures the model. It's useful to note that the f1-score weights precision and recall equally. I won't go further into this here, but it is an assumption that could be discussed.

Results from the classification reports			
	Precision	Recall	f1-score
Baseline model	0.64	0.64	0.64
Extended model	0.84	0.84	0.84

Figure 14 - Classification report results for the baseline and extended model

Figure 14 shows the extended model performed better than the baseline model in all three categories. As discussed in the data storytelling and inferential statistics section, the skewedness of all the variables and the fact that there are many classes that occur very rarely, means there are several obstacles in the data to improve the model further.

Testing with domain experts

In order to test the model, the domain experts at EFO tried the classification to see how the model worked, and what it predicted.

Overall, the accuracy of the model seemed to be about the same as from the classification report. For "normal" technical descriptions, this was impressive enough. For some of the more difficult and abbreviation-heavy technical descriptions, it was very impressive to see that the model correctly classified products, where many humans would not be able to even decipher the text.

The results from the testing gives a powerful insight: the classification model can also be used as a "search engine" of sorts. If one provides single words or terms that are expected to only show up in certain products, one can see if the model agrees. An example was the word "pet" in Norwegian. According to one of the domain experts, this should mostly or only be used for movement sensors for intrusion detection system where some products have a pet detection system. The model correctly classified this, to the satisfaction of the domain expert in question. For all the other "search" examples we tried, the model correctly or very nearly predicted the right class.

"Very nearly predicted" may need some more explanation. The ETIM-classes are created to match different types of products. An example could be a light fixture that hangs from the ceiling, such as a chandelier. Here, the model could predict a light fixture that is mounted on the ceiling but is not hanging. Though the difference is small, this is a misclassification.

A different example of classification error was if one searches using the word “dimmer”, where spot luminaire gets a slightly higher probability than dimmers. This could be because of the use of the word “dimmer” is more used in spot luminaire technical descriptions than the dimmers themselves for instance. It could also be because many of the ETIM-features (which are not discussed in this project) have the word “dimmer” in them, which could perhaps be used in the technical descriptions also. Although it’s wrongly classified, it’s predicts a product that is closely linked, which might be useful if one is trying to find related products.

Going back to the intended use of the classification model with the domain experts, about half of the times the model misclassified products, the errors were in the “nearly”-category. This is of course not as useful as a correct classification, but it could still be useful. The comment from the domain experts is that for the deployment of the model, one should display similar ETIM-classes, so that the model can actively help with correctly classifying products, even when it misclassifies.

Conclusions and future work

The final model was able to give good predictions. Both the results in the classification report and the results from the testing with the domain experts at EFO show that the predictions from this model can be used to help EFO’s members with classification to a reasonable degree of certainty.

In addition to giving the highest predicted class, one could show the top five highest predicted classes for instance. By showing more of what the model predicts, one might give more agency to the users, who feel like they have to make an informed decision based on help rather than trusting a machine learning model blindly.

When the model misses, it often is very close to the correct class – perhaps one could show similar product classes to the one the model predicted to alleviate some of these errors.

Both the baseline and the extended models are based on a single, static dataset. In the future, one might consider updating the dataset intermittently so that new data can help improve the predictions for the rarest classes.

There is also potential to do more with the skewedness inherent in the variables. This could be synthetic sampling or undersampling for instance. Either way, if one could increase products in the rarest ETIM-classes, that would go a long way towards getting a better model.

Recommendations

I recommend that EFO make the model available to their members, so that it can be used and tested in real life. With use, I would expect examples of extreme predictions to be brought forward, which can be helpful in the further development of the model.

EFO has other classification models than just ETIM. The foundations of this model can be used to quickly create predictions for those classifications with little extra work.