UNIVERSITY OF LONDON

## MSc Data Science

## Module: Statistics and Statistical Data Mining

## Task Name: Product Categorisation using Python 3

- Please Note: You are permitted to upload your Coursework in the final submission area as many times as you like before the deadline. You will receive a similarity/originality score which represents what the Turnitin system identifies as work similar to another source. The originality score can take over 24 hours to generate, especially at busy times e.g. submission deadline.

- If you upload the wrong version of your Coursework, you are able to upload the correct version of your Coursework via the same submission area. You simply need to click on the 'submit paper' button again and submit your new version before the deadline.

  In doing so, this will delete the previous version which you submitted and your new updated version will replace it. Therefore your Turnitin similarity score should not be affected. If there is a change in your Turnitin similarity score, it will be due to any changes you may have made to your Coursework.

- Please note, when the due date is reached, the version you have submitted last, will be considered as your final submission and it will be the version that is marked.

- **Once the due date has passed, it will not be possible for you to upload a different version of your assessment. Therefore you must ensure you have submitted the correct version of your assessment which you wish to be marked by the due date.**

**You are asked to submit a Jupyter notebook or Python script that contains your solution (Weighted at 50% of final mark for the module). Please make sure you use Python 3 and not Python 2. Python 2 code will not be marked and will be considered as a non-submission.**

## Coursework Description

**Task Name: Product Categorisation**
The correct categorisation of products is a very important step for online shops. This step might look straightforward, but it can easily be a nightmare. This could be due to any number of difficulties, including: 1) the number of available categories can be huge, 2) the number of available categories can change constantly, and 3) new products may be added daily.

Manual categorisation of products can be a tedious and labour-intensive task. Therefore, it makes sense to automate this process. One method is to use machine learning.

**In this task,** you will implement a product categoriser based on the following explanation. You are expected to learn some simple techniques that are required to finish this task (this is if you do not already know them).

**Here is a description of a good approach:**
As shown in the table below, there are three levels of categories. Level 1 is the highest and most generic and level 3 is the most specific. The idea is to create multiple models over a number of iterations.

In the first pass we create one model using the features and the Level 1 column as the class variable.
In the next pass (to predict level 2), we create one separate model for each unique category in Level 1.

Similarly, for Level 3, the number of models we create is the same as the number of distinct groups of combinations of the categories in Level 1 and Level 2.

| Description | Level_1 | Level_2 | Level_3 |
|---|---|---|---|
| gerb cap help keep littl on head cov warm day ... | 09BF5150 | C7E19 | D06E |
| newborn inf toddl boy hoody jacket oshkosh b g... | 2CEC27F1 | ADAD6 | 98CF |
| tut ballet anym leap foxy fash ruffl tul toddl... | 09BF5150 | C7E19 | D06E |
| newborn inf toddl boy hoody jacket oshkosh b g... | 2CEC27F1 | ADAD6 | 98CF |
| easy keep feel warm cozy inf toddl girl hoody ... | 2CEC27F1 | ADAD6 | 98CF |
| newborn inf toddl boy hoody jacket oshkosh b g... | 2CEC27F1 | ADAD6 | 98CF |
| mit warm protect real stay dainty littl hand p... | 09BF5150 | C7E19 | D06E |
| fal back cozy bas toughskin inf toddl girl mic... | 2CEC27F1 | ADAD6 | 98CF |
| ev smal lumberjack nee cozy look cool newborn ... | 2CEC27F1 | ADAD6 | 98CF |
| easy keep feel warm cozy inf toddl girl hoody ... | 2CEC27F1 | ADAD6 | 98CF |
| fal back cozy bas toughskin inf toddl girl mic... | 2CEC27F1 | ADAD6 | 98CF |
| get much ad toddl boy sherp hoody jacket boy r... | 2CEC27F1 | ADAD6 | 98CF |
| may seem lik riddl inf toddl boy hoody jacket ... | 2CEC27F1 | ADAD6 | 98CF |
| cold hand wint thank three pair toddl girl mit... | 09BF5150 | C7E19 | D06E |
| effortless styl start ad inf girl littl wond h... | 2CEC27F1 | ADAD6 | 98CF |

The total number of models we need to create depends on the number of categories. For example, if the number of unique categories was 12, 22 and 31 in Levels 1, 2 and 3 respectively, then the total number of models created will be one model in the first pass, 12 models in the second pass and 22 models in the third pass. This makes the total number of models 35. In a mock run of a model solution to this task, the number of models was between 50 and 60. The number you will have can be different as it depends on how you split the data into train/test splits.

**Data Preprocessing:**

You will have access to a dataset that contains product descriptions and categories (i.e. each product will belong to three levels of a hierarchy). The data has been prepared to be suitable for this task.

The data might contain missing values in some of its columns so we will leave it to you to explore it and handle it (e.g. you can drop instances that contain one or more missing values).

In addition, as the data contains three label columns (i.e. Level_1, Level_2 and Level_3), the number of instances in each class (in each column) can be

really small. As a preprocessing step, you are asked to remove classes where the number of instances is less than 10. You need to check the three label columns (i.e. Level_1, Level_2 and Level_3) and if any class in any level contains less than 10 instances then you should remove those instances.

**Feature extraction:**
You will have noticed that in the description column of the table, the description is in textual format which is not suitable for machine learning as is (in other words, we need to extract numeric features so that we can use them to train and test classifiers). As part of this task, you are expected to transform the textual description into a numeric feature matrix. One simple technique to do that is the **TF-IDF** (*term frequency-inverse document frequency*) method which is straightforward to use, or apply, in Python using the sci-kit learn package. Please observe that the resulting feature matrix can be large. Feel free to use a subset of it if it does not fit into your computer's memory. However, before TF-IDF can be applied, you will need to clean up and prepare the textual data correctly. Here is a list of steps that you can apply to the description column before applying TF-IDF:

a- Convert text to lowercase.
b- Remove punctuation marks.
c- Apply stemming using the popular Snowball or Porter Stemmer.
d- Apply NGram Tokenization.

**Which classifier to use:**
To make this task flexible, we are going to allow you to use your favourite classifier.

**Evaluation and Future Predictions:**
During the model training process, you will need to save each model you create (you can use Python pickle). For evaluation and future predictions, the first step would be to load the level 1 model and predict the level 1 category. Based on this prediction, you load the appropriate level 2 model and predict the level 2 category. And based on this prediction, you load the appropriate level 3 model and predict the level 3 category. Please observe that a large number of models might be generated which can make training and prediction take several minutes.
For a performance evaluation metric, use accuracy as your metric. You should generate a separate accuracy value for each level.

## Assessment Criteria:

This coursework makes 50% of the total module, and it is worth 50 marks. Here is a breakdown of steps required to finish this task and the marks for each step:

1- Load data and perform exploratory analysis (4 marks).
2- Inspect data for missing values and successfully deal with them (4 marks).
3- Drop classes where the number of instances is < 10 (4 marks).
4- Prepare text for TF-IDF as explained in the task description (4 marks).
5- Apply TF-IDF and extract a feature matrix (10 marks).
6- Split data into train and test splits (4 marks).
7- Train models for the three levels as explained in the task description (8 marks).
8- Predict the test set as explained in the task description (8 marks).
9- Based on the predictions and actual value of each level, compute accuracy value of each level (4 marks).

Please refer to Appendix C of the Programme Regulations for detailed Assessment Criteria.

## Plagiarism:

This is cheating. Do not be tempted and certainly do not succumb to temptation. Plagiarised copies are invariably rooted out and severe penalties apply. All assignment submissions are electronically tested for plagiarism.