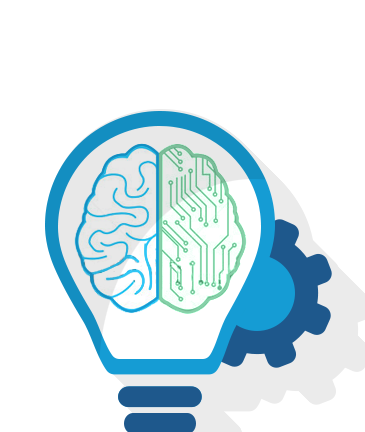
Business Intelligence for Managers

Project Report

**On**

**PRODUCT CATEGORIZATION IN AN**

**E-COMMERCE CATALOG**



**PRODUCT CATEGORIZATION IN AN E-COMMERCE CATALOG**

1. **Introduction**

Online stores and E-commerce websites have a list of products listed online. The organizations and businesses operating these online stores and websites know that if people are not able to locate products in the large plethora of their product listings will result in the loss of revenue and a bad user experience. We here present a solution to improve this major problem that online stores and e-commerce websites face while listing their products online.

**What is Product Categorization?**

It can be defined as the process that divides the products listed online into groups. Generally this uses a three tier architecture. You have a superclass, a category and a sub category. Let’s take an example to explain the same. I am a customer who needs to purchase a mobile case. Now to search for mobile cases I need to go to the mobiles section, which is the category of the product I am trying to search. The mobile category comes under the Electronics superclass. So my superclass, category and sub category in this case are Electronics, Mobiles and Mobile cases respectively.

**What are the benefits of product categorizations?**

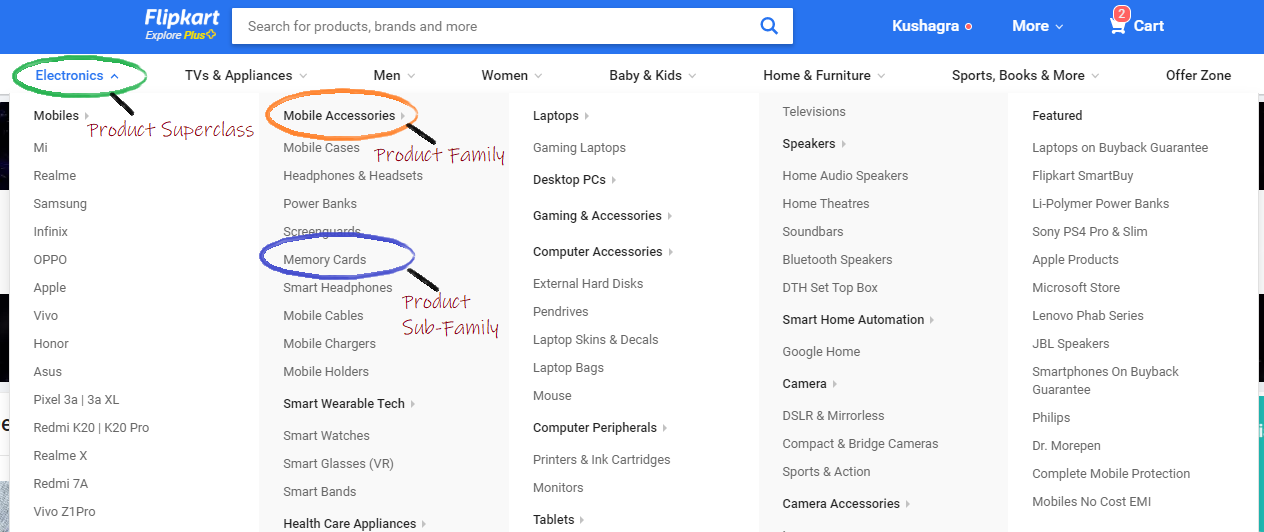
A very well structured product categorization helps to design an effective and efficient system which provides the following benefits:

1. Improved Reporting: Right product categorization helps in the right reporting of the product catalogue and in generating the reports like sales summary for the category. Wrong listed products will influence these reports and alter the figures generated.
2. Better searching: If the products are listed in the right category their searching becomes easy as most often people search products based on the hierarchy.
3. Improved Workflow: Internal people in the organization can have an improved workflow for the category as mis classification is reduced.
4. Increased Revenue : Since the products are rightly classified and are listed in their respective categories the chances of their purchase are increased which will eventually increase in revenue.
5. Better User Experience: In case of wrong categorization products will be visible in the wrong category which will result in confusion from client side. This will result in bad user experience.
6. Duplicate Products Detection and Removal: Products that are listed under different names from the same seller can be detected. Other duplicates can also be detected and removed.

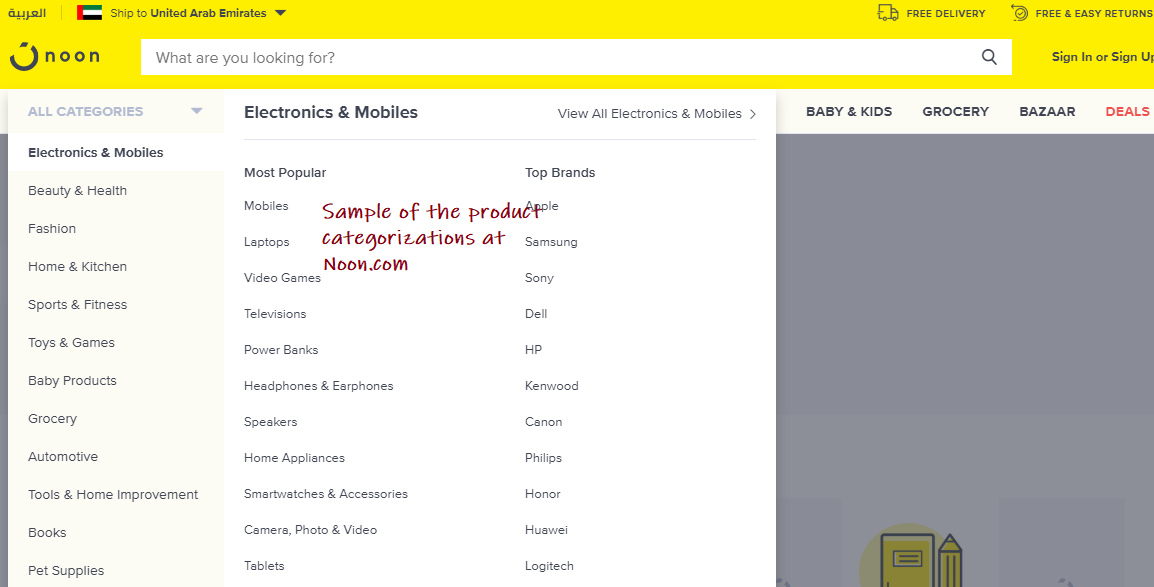
In our case we have taken the data of Noon.com. It’s an e-commerce website based out of Dubai. We shall discuss the process that is involved in product categorizations and what are the implications that arise if the right categorizations are not done. Noon has minimum defined attributes that help it correctly and efficiently categorize the products. These attributes can be anything that helps define the product i.e. in case of a mobile phone they can be the RAM, battery and the OS of the product in question. If the minimum defining attributes of a product listed do not match the category that they are listed, they are not displayed in the specific category. This results in the loss of revenue due to the product not being available to the public and also results in a bad user experience as the right classification has not been done.

We have attached an image below to highlight the bifurcations for **Flipkart.com** :

It shows the superclass, category and sub category of a product that one might want to buy.



Similarly we have an image of the **Noon.com** website whose data that we are working upon.



1. **Description of data**

**Metadata (Data about data)**

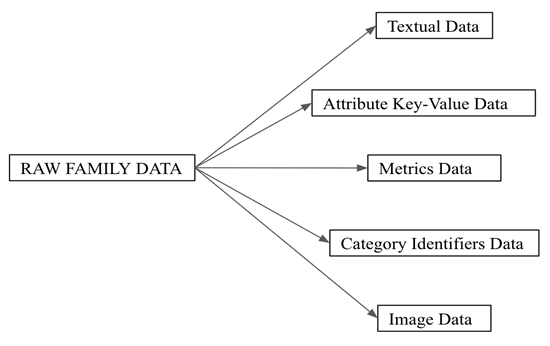
In an e-commerce company, there is a need for highly structured and organized data. So at Noon, data is spread across multiple databases inside multiple tables or schemas. Each of these tables holds different kinds of information like, text, attributes, features, metadata about attributes, offers on a product, pricing information, quantity in stock, completeness status (which depicts that how much is product ready so that it can be taken live), active status, image status, seller information, brand data, and much more.

* Every product is classified at 3 levels, first is the product category, second is the product type, and third is product subtype.
* This classification helps in clustering the products belonging to the same sub-category which is useful while displaying the search results as well as give better control over the data.
* There are numerous predefined attributes based on product category type which are used to represent the product. These predefined attributes are further refined based on the product type, and finally, these products are refined on the basis of attributes that are necessary for their subtype.

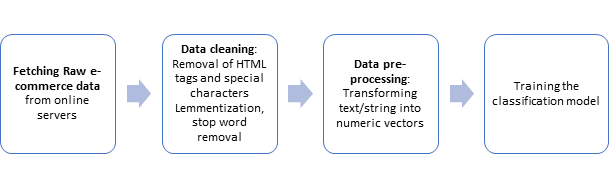
**Nature of data**

There are 5 types of data belonging to every product.

* The first type of data that is common for every product is textual data, which includes information like, title, description and feature bullets. This type of data majorly consists of pure English text along with some product specific information like model number and size.
* The second type of data is the Category Identifiers which are product’s category (like “baby care”), type (like “baby bath”), and subtype (like “baby soap”, “baby shampoo”). They are also called as full-type identifiers. The values filled in these attributes are integers which are the keys of the dictionaries defined for category, type, and subtype.
* The third type of data is the metrics and dimension data like length, width, height, weight, RAM, Processor Speed, Camera Resolution. In short, these attributes contain information about the product which is quantifiable and has some S.I. unit.
* The fourth type of data is the attribute key-value pairs. These attributes are generally in the form of dictionary, in which both the keys as well as values can contain either short textual data or some numerical data or some metric data. These types of attributes are not standard and may vary from seller to seller.



1. **Methodology used**

****

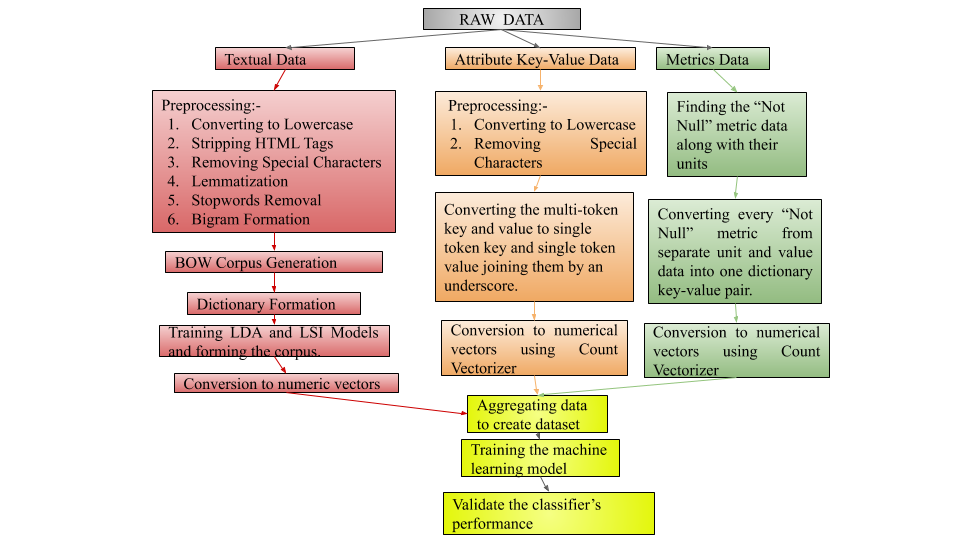
Tools and Technologies Used

a) **Gensim**: Gensim is a free Python library designed to extract semantic topics from documents, as efficiently (computer-wise) and painlessly (human-wise) as possible. Gensim is designed to process raw, unstructured digital texts (“plain text”). The algorithms in Gensim, such as Word2Vec, FastText, Latent Semantic Analysis (LSI, LSA), Latent Dirichlet Allocation (LDA) etc., discover the semantic structure of documents by examining statistical co-occurrence patterns within a corpus of training documents. These algorithms are unsupervised.

b) **NLTK**: The Natural Language Toolkit (NLTK) is a Python package for natural language processing. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning and wrappers for industrial-strength NLP libraries.

c) Python

For each category, train data is prepared and stored. The following steps describe how the training data is prepared.



3.1. String/textual data

At first, the whole data is divided on the basis of the product category. Pre-processing of the text is the most important step. It helps in removing the noisy data which can lead to poor results.

Pre-processing involves the following steps:

a) Converting the whole text to lowercase

b) Stripping HTML tags

c) Lemmatization

d) Replacing special characters by blank space - Special characters are not required while product matching. For example, if two products have titles as “Samsung Galaxy-A8” and “Samsung Galaxy A8” respectively. Then the title of these two products will not be found similar if the special character ‘-’ is not replaced by blank space.

e) Removing Stop Words

f) Forming bigrams and phrases - Bigrams are formed by replacing the blank space between the two frequently occurring words by an underscore character. These results in the combining of two tokens to a single token. For example, without these steps “New York” will be treated as two different tokens “New” and “York” while training. After this step, they will be treated as a single token like “New\_York”

After pre-processing, the whole text is tokenized, and a dictionary is created. Using the pre-processed corpus and the dictionary, a BOW (Bag of Words) representation of whole training corpus is prepared. This representation is called the BOW corpus. This BOW corpus is fed to the Gensim LDA module. The Gensim LDA module extracts topics from this corpus and creates a new corpus. The new corpus is called the LDA corpus and it includes the topics and features learned by the model. The LDA corpus is formed for each category and is stored, which can be further used for index calculation and evaluation of the query product. A similar approach is followed with LSI model.

3.2. Attribute Key-Value Pairs

Following are the steps to pre-process Attribute Key-Value Pairs:

a) Converting the data to lowercase.

b) Removing the special characters.

c) Converting the text in the key-value pairs

While training for each category, a list of processed attributes for each product in the category is prepared and saved.

3.3. Metrics and Dimensions Data

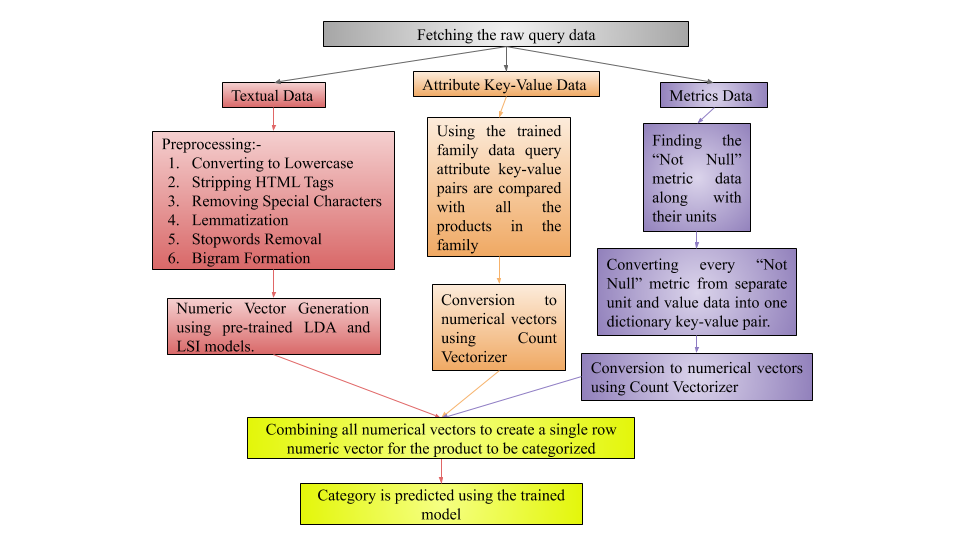
For preparing the data for metrics, we created a list of dictionaries, in which each dictionary holds data like {‘product\_length’:[100.0, ‘cm’]}. Such a key-value pair is generated from raw data that looks like as follows {‘product\_length’:100.0,‘product\_length\_unit’: ‘cm’}.

After cleaning and pre-processing the text/string data into numeric vectors, the machine learning model was trained and then validated for its performance.

1. **Analysis**

All the analysis has been done in Python and following libraries have been used:

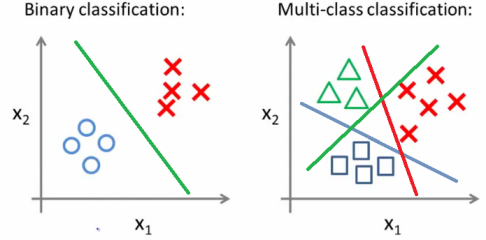
1. NLTK - The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.
2. Gensim - Gensim is an open-source library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. Gensim is implemented in Python and Cython.
3. Sklearn - Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
4. Numpy - NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
5. Pandas - pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
6. Seaborn - Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.



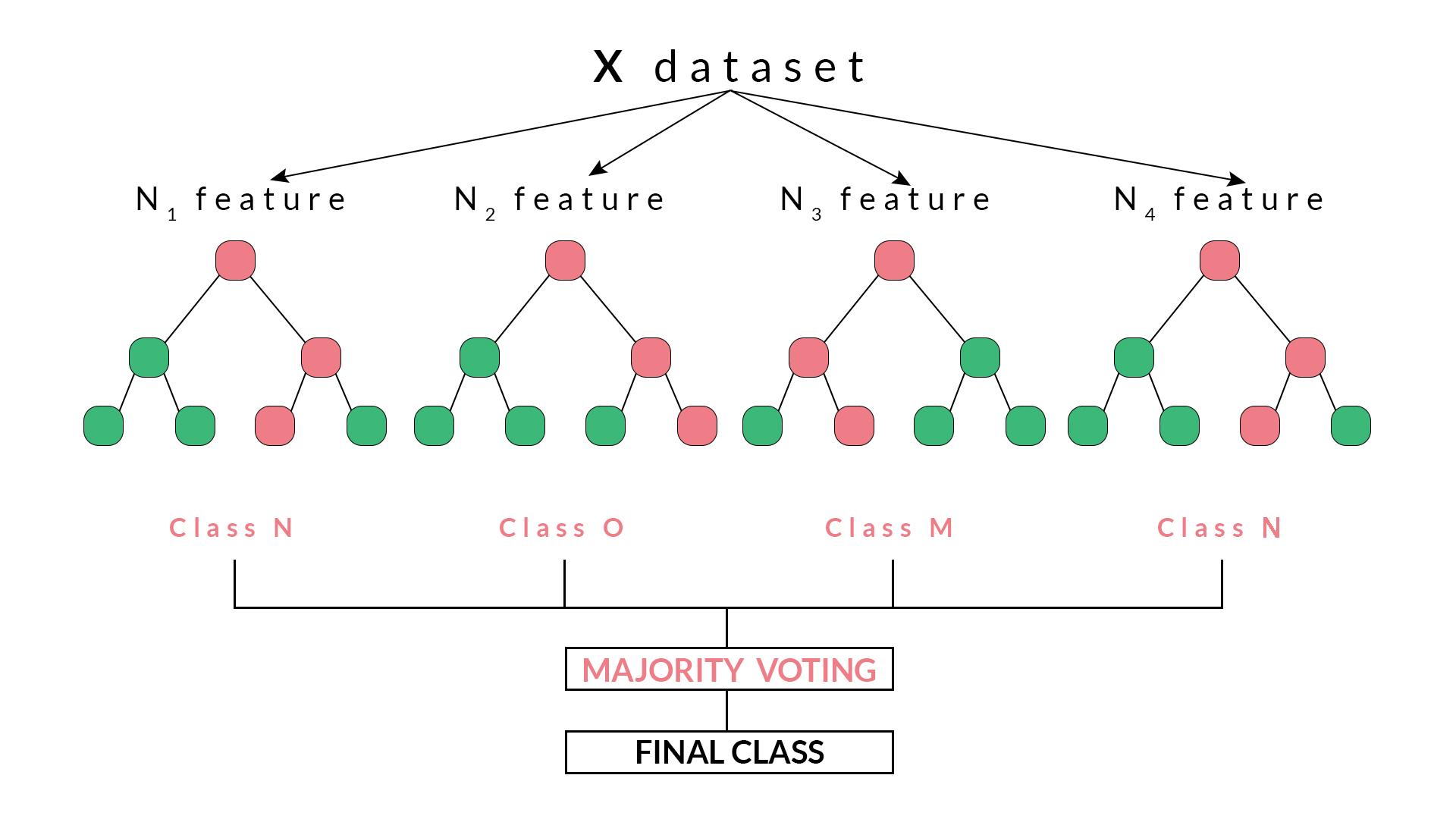
The titles are converted to vectors using Latent Semantic Analysis (LSI). For the description of the product, the Latent Dirichlet Allocation is used to convert them into numerical vectors and for Attribute-Value pairs, Count Vectorizer is used.

We combine the LSI features, LDA features and Count Vectorizer features for every product for forming the final data which will be used for training the model. The final data contains approximately 95,000 samples with 1,76,216 features/columns belonging to 190 classes.

After analyzing the data in context to the problem we recognized this as a supervised machine learning problem. More specifically it is a multi-class classification problem. It is a classification task with more than two classes; e.g., classify a set of images of fruits which may be oranges, apples, or pears. Multi-class classification makes the assumption that each sample is assigned to one and only one label: a fruit can be either an apple or a pear but not both at the same time.



We have used the Random Forest for classification. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.



The major reason for using Random Forest is that it has the following advantages:

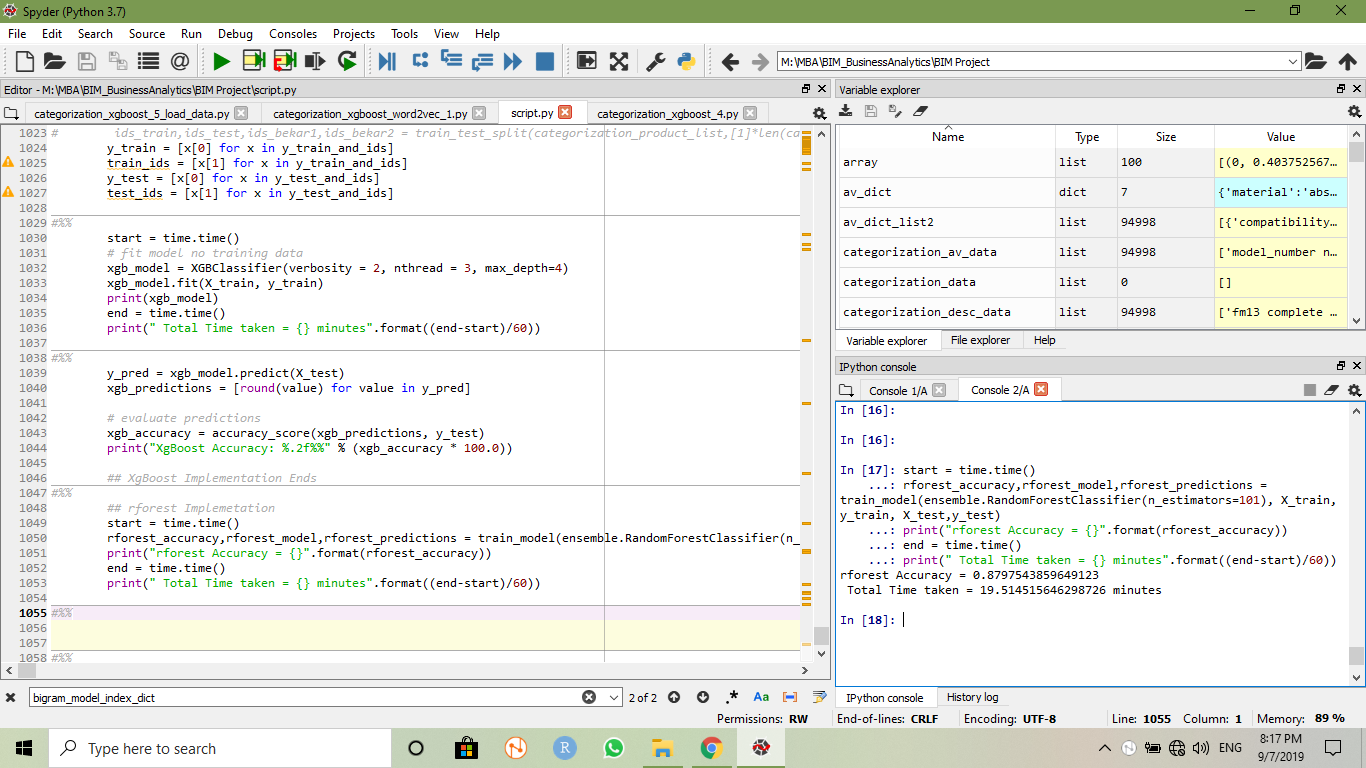
* A single decision tree tends to overfit the data. The process of averaging or combining the results of different decision trees helps to overcome the problem of overfitting.
* Random forests also have less variance than a single decision tree. It means that it works correctly for a large range of data items than single decision trees.
* Random forests are extremely flexible and have very high accuracy.
* They also do not require preparation of the input data. You do not have to scale the data.
* It also maintains accuracy even when a large proportion of the data are missing.

The training data is segregated into independent and dependent variables and then is split into training and testing set. The data is divided into training and validation set equally. It means that the size of the training data is 50% and the size of the validation set is also 50% of the total data. The number of estimators or the decision trees to be made while training are set to be 101.

The training dataset of the size (47500 X 176216) is fed to Random Forest and training is done. It takes approximately 19 minutes to train the whole model. The data is then tested against the whole validation set.

1. **Results**

**The accuracy of the model is around 88%.**

****

**The rounded up metrics of the predicted categories is as follows:**

1. Mean precision = 0.868560606849865

The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

1. Mean recall = 0.8680922220681879

The recall is the ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.

1. Mean f1-score = 0.8670250735510946

The F-beta score can be interpreted as a weighted harmonic mean of precision and recall, where an F-beta score reaches its best value at 1 and worst score at 0. The F-beta score weights recall more than precision by a factor of beta. beta == 1.0 means recall and precision are equally important.

1. Mean support = 250.0

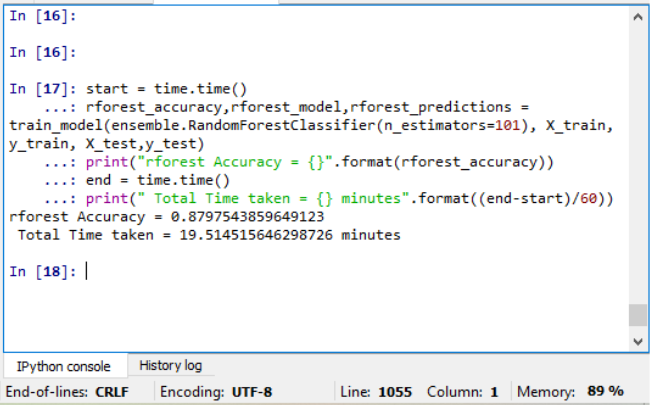
The support is the number of occurrences of each class in y\_true.

Following is the detailed report of classification done on validation set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classes | precision | recall | f1-score | support |
| 0 | 0.916667 | 0.916667 | 0.916667 | 252 |
| 1 | 0.8 | 0.770492 | 0.784969 | 244 |
| 2 | 0.76834 | 0.799197 | 0.783465 | 249 |
| 3 | 0.95102 | 0.947154 | 0.949084 | 246 |
| 4 | 0.807843 | 0.85124 | 0.828974 | 242 |
| 5 | 0.838951 | 0.868217 | 0.853333 | 258 |
| 6 | 0.754647 | 0.831967 | 0.791423 | 244 |
| 7 | 0.980545 | 0.947368 | 0.963671 | 266 |
| 8 | 0.946768 | 0.988095 | 0.96699 | 252 |
| 9 | 0.930657 | 0.969582 | 0.949721 | 263 |
| 10 | 0.964286 | 0.968127 | 0.966203 | 251 |
| 11 | 0.840741 | 0.900794 | 0.869732 | 252 |
| 12 | 0.894545 | 0.960938 | 0.926554 | 256 |
| 13 | 0.852535 | 0.777311 | 0.813187 | 238 |
| 14 | 0.85259 | 0.85259 | 0.85259 | 251 |
| 15 | 0.821862 | 0.86383 | 0.842324 | 235 |
| 16 | 0.888031 | 0.905512 | 0.896686 | 254 |
| 17 | 0.799213 | 0.812 | 0.805556 | 250 |
| 18 | 0.972441 | 0.980159 | 0.976285 | 252 |
| 19 | 0.964912 | 0.93617 | 0.950324 | 235 |
| 20 | 0.976834 | 0.992157 | 0.984436 | 255 |
| 21 | 0.992218 | 0.984556 | 0.988372 | 259 |
| 22 | 0.97561 | 0.991736 | 0.983607 | 242 |
| 23 | 0.98 | 0.976096 | 0.978044 | 251 |
| 24 | 0.732075 | 0.811715 | 0.769841 | 239 |
| 25 | 0.842697 | 0.914634 | 0.877193 | 246 |
| 26 | 0.842342 | 0.724806 | 0.779167 | 258 |
| 27 | 0.801556 | 0.847737 | 0.824 | 243 |
| 28 | 0.852535 | 0.829596 | 0.840909 | 223 |
| 29 | 0.913242 | 0.757576 | 0.828157 | 264 |
| 30 | 0.918367 | 0.957447 | 0.9375 | 235 |
| 31 | 0.849421 | 0.827068 | 0.838095 | 266 |
| 32 | 0.864 | 0.874494 | 0.869215 | 247 |
| 33 | 0.957031 | 0.949612 | 0.953307 | 258 |
| 34 | 0.941909 | 0.953782 | 0.947808 | 238 |
| 35 | 0.812977 | 0.872951 | 0.841897 | 244 |
| 36 | 0.951965 | 0.923729 | 0.937634 | 236 |
| 37 | 0.896996 | 0.86722 | 0.881857 | 241 |
| 38 | 0.863281 | 0.856589 | 0.859922 | 258 |
| 39 | 0.848606 | 0.832031 | 0.840237 | 256 |
| 40 | 0.987552 | 1 | 0.993737 | 238 |
| 41 | 0.816479 | 0.810409 | 0.813433 | 269 |
| 42 | 0.813187 | 0.874016 | 0.842505 | 254 |
| 43 | 0.904959 | 0.855469 | 0.879518 | 256 |
| 44 | 0.69863 | 0.593023 | 0.641509 | 258 |
| 45 | 0.762887 | 0.880952 | 0.81768 | 252 |
| 46 | 0.673171 | 0.570248 | 0.61745 | 242 |
| 47 | 0.832685 | 0.89916 | 0.864646 | 238 |
| 48 | 0.729242 | 0.779923 | 0.753731 | 259 |
| 49 | 0.728873 | 0.805447 | 0.76525 | 257 |
| 50 | 0.790476 | 0.648438 | 0.712446 | 256 |
| 51 | 0.934694 | 0.942387 | 0.938525 | 243 |
| 52 | 0.829787 | 0.966942 | 0.89313 | 242 |
| 53 | 0.920863 | 0.951673 | 0.936015 | 269 |
| 54 | 0.992157 | 0.996063 | 0.994106 | 254 |
| 55 | 0.782609 | 0.874494 | 0.826004 | 247 |
| 56 | 0.885714 | 0.815789 | 0.849315 | 266 |
| 57 | 0.928571 | 0.736842 | 0.82167 | 247 |
| 58 | 0.682008 | 0.654618 | 0.668033 | 249 |
| 59 | 0.867647 | 0.92549 | 0.895636 | 255 |
| 60 | 0.925439 | 0.764493 | 0.837302 | 276 |
| 61 | 0.654472 | 0.631373 | 0.642715 | 255 |
| 62 | 0.85098 | 0.789091 | 0.818868 | 275 |
| 63 | 0.980916 | 0.969811 | 0.975332 | 265 |
| 64 | 0.915058 | 0.925781 | 0.920388 | 256 |
| 65 | 0.917603 | 0.953307 | 0.935115 | 257 |
| 66 | 0.979424 | 0.940711 | 0.959677 | 253 |
| 67 | 0.926531 | 0.986957 | 0.955789 | 230 |
| 68 | 0.959016 | 0.943548 | 0.95122 | 248 |
| 69 | 0.967273 | 0.985185 | 0.976147 | 270 |
| 70 | 0.735178 | 0.775 | 0.754564 | 240 |
| 71 | 0.841004 | 0.817073 | 0.828866 | 246 |
| 72 | 0.782051 | 0.7625 | 0.772152 | 240 |
| 73 | 0.850746 | 0.919355 | 0.883721 | 248 |
| 74 | 0.987552 | 0.944444 | 0.965517 | 252 |
| 75 | 0.960474 | 0.995902 | 0.977867 | 244 |
| 76 | 1 | 1 | 1 | 253 |
| 77 | 0.879167 | 0.840637 | 0.85947 | 251 |
| 78 | 0.877049 | 0.816794 | 0.84585 | 262 |
| 79 | 0.937255 | 0.987603 | 0.961771 | 242 |
| 80 | 0.972441 | 0.961089 | 0.966732 | 257 |
| 81 | 0.926471 | 0.976744 | 0.950943 | 258 |
| 82 | 0.988506 | 0.988506 | 0.988506 | 261 |
| 83 | 0.984556 | 1 | 0.992218 | 255 |
| 84 | 0.96875 | 0.995984 | 0.982178 | 249 |
| 85 | 0.934343 | 0.852535 | 0.891566 | 217 |
| 86 | 0.80292 | 0.873016 | 0.836502 | 252 |
| 87 | 0.968127 | 0.960474 | 0.964286 | 253 |
| 88 | 0.93254 | 0.886792 | 0.909091 | 265 |
| 89 | 0.660714 | 0.764463 | 0.708812 | 242 |
| 90 | 0.584746 | 0.554217 | 0.569072 | 249 |
| 91 | 0.772563 | 0.823077 | 0.79702 | 260 |
| 92 | 0.760504 | 0.693487 | 0.725451 | 261 |
| 93 | 0.864151 | 0.938525 | 0.899804 | 244 |
| 94 | 0.882576 | 0.94332 | 0.911937 | 247 |
| 95 | 0.740741 | 0.711462 | 0.725806 | 253 |
| 96 | 0.933884 | 0.91498 | 0.924335 | 247 |
| 97 | 0.876984 | 0.88755 | 0.882236 | 249 |
| 98 | 0.882927 | 0.741803 | 0.806236 | 244 |
| 99 | 0.937759 | 0.945607 | 0.941667 | 239 |
| 100 | 0.951417 | 0.979167 | 0.965092 | 240 |
| 101 | 0.961977 | 0.973077 | 0.967495 | 260 |
| 102 | 0.981273 | 0.945848 | 0.963235 | 277 |
| 103 | 0.988 | 0.988 | 0.988 | 250 |
| 104 | 0.922449 | 0.933884 | 0.928131 | 242 |
| 105 | 0.964 | 0.964 | 0.964 | 250 |
| 106 | 1 | 0.971545 | 0.985567 | 246 |
| 107 | 0.975709 | 0.99177 | 0.983673 | 243 |
| 108 | 0.798246 | 0.686792 | 0.738337 | 265 |
| 109 | 0.909804 | 0.966667 | 0.937374 | 240 |
| 110 | 0.846154 | 0.965649 | 0.901961 | 262 |
| 111 | 0.973783 | 0.977444 | 0.97561 | 266 |
| 112 | 0.952941 | 0.938224 | 0.945525 | 259 |
| 113 | 0.985185 | 0.996255 | 0.990689 | 267 |
| 114 | 0.99187 | 0.995918 | 0.99389 | 245 |
| 115 | 0.65625 | 0.691358 | 0.673347 | 243 |
| 116 | 0.717742 | 0.726531 | 0.72211 | 245 |
| 117 | 0.854701 | 0.766284 | 0.808081 | 261 |
| 118 | 0.963415 | 0.979339 | 0.971311 | 242 |
| 119 | 0.988095 | 0.972656 | 0.980315 | 256 |
| 120 | 0.983806 | 0.987805 | 0.985801 | 246 |
| 121 | 0.853383 | 0.869732 | 0.86148 | 261 |
| 122 | 0.85124 | 0.81746 | 0.834008 | 252 |
| 123 | 0.964706 | 0.960938 | 0.962818 | 256 |
| 124 | 0.913534 | 0.956693 | 0.934615 | 254 |
| 125 | 0.972112 | 0.938462 | 0.95499 | 260 |
| 126 | 0.852518 | 0.936759 | 0.892655 | 253 |
| 127 | 0.917391 | 0.864754 | 0.890295 | 244 |
| 128 | 0.988327 | 0.996078 | 0.992188 | 255 |
| 129 | 0.827731 | 0.781746 | 0.804082 | 252 |
| 130 | 0.881533 | 0.947566 | 0.913357 | 267 |
| 131 | 0.692913 | 0.671756 | 0.682171 | 262 |
| 132 | 0.785455 | 0.915254 | 0.845401 | 236 |
| 133 | 0.943396 | 0.77821 | 0.852878 | 257 |
| 134 | 0.996032 | 0.992095 | 0.994059 | 253 |
| 135 | 0.975806 | 0.979757 | 0.977778 | 247 |
| 136 | 0.966667 | 0.978903 | 0.972746 | 237 |
| 137 | 0.796791 | 0.613169 | 0.693023 | 243 |
| 138 | 0.898396 | 0.714894 | 0.796209 | 235 |
| 139 | 0.97541 | 0.959677 | 0.96748 | 248 |
| 140 | 0.722944 | 0.676113 | 0.698745 | 247 |
| 141 | 0.768116 | 0.80916 | 0.788104 | 262 |
| 142 | 0.768182 | 0.704167 | 0.734783 | 240 |
| 143 | 0.78673 | 0.653543 | 0.713978 | 254 |
| 144 | 0.638796 | 0.782787 | 0.703499 | 244 |
| 145 | 0.67907 | 0.584 | 0.627957 | 250 |
| 146 | 0.934426 | 0.930612 | 0.932515 | 245 |
| 147 | 0.756 | 0.825328 | 0.789144 | 229 |
| 148 | 0.892857 | 0.819672 | 0.854701 | 244 |
| 149 | 0.457912 | 0.569038 | 0.507463 | 239 |
| 150 | 0.815385 | 0.812261 | 0.81382 | 261 |
| 151 | 0.960474 | 0.927481 | 0.943689 | 262 |
| 152 | 0.843137 | 0.947137 | 0.892116 | 227 |
| 153 | 0.770751 | 0.789474 | 0.78 | 247 |
| 154 | 0.78 | 0.730337 | 0.754352 | 267 |
| 155 | 0.879167 | 0.864754 | 0.871901 | 244 |
| 156 | 0.704225 | 0.595238 | 0.645161 | 252 |
| 157 | 0.779167 | 0.809524 | 0.794055 | 231 |
| 158 | 0.936975 | 0.936975 | 0.936975 | 238 |
| 159 | 0.842795 | 0.784553 | 0.812632 | 246 |
| 160 | 0.834586 | 0.956897 | 0.891566 | 232 |
| 161 | 0.988 | 1 | 0.993964 | 247 |
| 162 | 0.917293 | 0.97992 | 0.947573 | 249 |
| 163 | 0.913043 | 0.916667 | 0.914851 | 252 |
| 164 | 0.940171 | 0.887097 | 0.912863 | 248 |
| 165 | 0.816143 | 0.781116 | 0.798246 | 233 |
| 166 | 0.765734 | 0.862205 | 0.811111 | 254 |
| 167 | 0.877637 | 0.742857 | 0.804642 | 280 |
| 168 | 0.840426 | 0.67234 | 0.747045 | 235 |
| 169 | 0.842308 | 0.948052 | 0.892057 | 231 |
| 170 | 0.884 | 0.936441 | 0.909465 | 236 |
| 171 | 0.79771 | 0.80695 | 0.802303 | 259 |
| 172 | 0.724138 | 0.771429 | 0.747036 | 245 |
| 173 | 0.812766 | 0.764 | 0.787629 | 250 |
| 174 | 0.888031 | 0.867925 | 0.877863 | 265 |
| 175 | 0.90625 | 0.845833 | 0.875 | 240 |
| 176 | 0.969811 | 0.958955 | 0.964353 | 268 |
| 177 | 0.934959 | 0.974576 | 0.954357 | 236 |
| 178 | 0.953488 | 0.964706 | 0.959064 | 255 |
| 179 | 0.909465 | 0.880478 | 0.894737 | 251 |
| 180 | 0.704453 | 0.728033 | 0.716049 | 239 |
| 181 | 0.757353 | 0.847737 | 0.8 | 243 |
| 182 | 0.874477 | 0.853061 | 0.863636 | 245 |
| 183 | 0.95102 | 0.924603 | 0.937626 | 252 |
| 184 | 0.820225 | 0.893878 | 0.855469 | 245 |
| 185 | 0.9375 | 0.90566 | 0.921305 | 265 |
| 186 | 0.701149 | 0.778723 | 0.737903 | 235 |
| 187 | 0.7 | 0.761506 | 0.729459 | 239 |
| 188 | 0.832714 | 0.896 | 0.863198 | 250 |
| 189 | 0.903361 | 0.86 | 0.881148 | 250 |
| avg / total | 0.869433 | 0.868105 | 0.867478 | 47500 |

1. **Interpretation and Outcomes of the results**

Results of the algorithm are shown in the image below:



**Interpretation and Outcomes:**

The random forest algorithm that was run gave an accuracy of 87.9% and took a total time of 19.51 minutes. With this range and length of data the accuracy received from running the algorithm was very good and completed in just 19.51 minutes. We had around 190 classes with each class having an approximate 500 products. So the total data was around 10000.

There have been organizations built that have been tackling these problems. This is a problem that forms a major roadblock for these e-commerce websites.

The outcome of this algorithm will result in:

1. **Proper categorization of every product in catalog**: The random forest algorithm will result in the right classification of products in their respective category. This will result in an increase in revenue, better product mapping and an enhanced user experience.
2. **Solidifying the minimum defining attributes for each category**: The minimum defining attributes that are used to define the product category will provide a solid solution for the future of the company. If these attributes can be used for right classification of each and every product i.e. we can receive a 100% accuracy these will solidify the place of the attributes for future use.
3. **Duplicate detection using minimum defining attributes**: If defined for each product the attributes should be able to rightly classify the products as well help in the detection of duplicates that have been added by different sellers to increase the selling of the product.
4. **Search results improvement**: When placed in the right category the products will ease out the searching process. Customer generally have a hierarchy defined in their heads and when mapped right this helps in making the searching process easy.

1. **Preventing the seller from tricking the system to capture buy box and doing justice to every seller**: Many sellers use multiple names and very small changes to list a product many times on a certain website. This results in people buying the product from the same sellers thus increasing their revenue. If the minimum defining attributes are used the right way, this will help in reducing the redundancies and reducing fake or multiple listings of the same products.
2. **Enhancing the user experience**: Right product mapping will enhance the user experience as customers will not have to face wrong products in the category they are looking for.
3. **Finally improving the company image and increasing the revenue**: If people are satisfied with the products and the listings this will increase the revenue and image of the company in the eyes of the consumers.
4. **References**

Following are some links which are used for technical support and reference throughout the project:-

1. Python Official Website - <https://www.python.org/>
2. Gensim LDA Model - <https://radimrehurek.com/gensim/models/ldamodel.html>
3. Gensim HDP Model - <https://radimrehurek.com/gensim/models/hdpmodel.html>
4. Gensim LSI Model - <https://radimrehurek.com/gensim/models/lsimodel.html>
5. NLTK Official Website - <http://www.nltk.org/>
6. <https://stackoverflow.com/questions/19233771/sklearn-plot-confusion-matrix-with-labels/31720054>
7. <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html>