Capstone Project Report

Machine Learning Engineer Nanodegree

Media Product Classification

Nagaraju Budigam
November 28th, 2017

Project Overview

Note: This is the original problem that I am developing at production level. Due to company compliance original datasets were not exposed, the provided datasets are resembling the original datasets.

Indix hosts the world's largest collection of programmatically accessible structured product information in the cloud. The products in our database belong to 25 verticals and that translates approximately to 6000 sub-categories. Every product that we carry in our database gets stamped with information about the "category" it belongs to.

Problem Statement

Aim of this project is to classifying a product into a particular category is very important to serve various use cases – like, helping search, performing product matching, providing category specific insights, and so on. The problem of stamping every product in our catalogue into a category is a **Multi Label Hierarchical Classification** problem. In this submission I have built a micro version of this classifier where I will predict 4 classes.

Evaluation Criteria

I would like to use accuracy as a evaluation metric to access the performance of the classifier.

- ➢ The evaluation metric would be accuracy score on the test set
    - o **Accuracy** is a common metric for binary classifiers; it takes into account both true positives and true negatives with equal weight.
    - o accuracy= true positives+ true negatives/dataset size.
- ➢ **predicted_output.csv** is the final predicted output file, which contains the test set with an additional column called **label** which that will have the predicted label of the product.
- ➢ There are 442041 records in the test set all of which would need a prediction

Analysis

Data Exploration

The datasets are available at the following locations. Kindly download and use.

Training data:

- https://drive.google.com/open?id=1q83Iyu3zrYf1JefwcaX2izpt7fp_xIiP

Test data:

- https://drive.google.com/file/d/1LlXRerCMWTDl6r1aeS9Cy960C7knYHz3/view?usp=sharing

Training data has the following fields and contains 603201 records

1. **storeId** - a unique number for identifying a website
2. **additionalAttributes** - Product attribute related to a particular product. These are key, value pairs that can be found in tabular format as product information for most products in e-commerce websites.

An example of **additionalAttributes**
{"ASIN": " B000JJRY9M",
"Amazon Bestsellers Rank": " in DVD & Blu-ray (See Top 100 in DVD & Blu-ray)",
"Average Customer Review": " Be the first to review this item",
"Classification": " Exempt",
"DVD Release Date": " 26 Feb. 2007",
"Format": " AC-3, Colour, Dolby, DVD-Video, PAL",
"Language": " English",
"Number of discs": " 1",
"Region": " Region 2 (This DVD may not be viewable outside Europe. Read more about DVD formats.)",
"Run Time": " 287 minutes",
"Studio": " Hip-O Records"}

3. **breadcrumbs**- breadcrumb captured at the page. Breadcrumbs typically appear horizontally across the top of a Web page, often below title bars or headers.

An example of breadcrumb
subjects > travel > world regions > europe > european nations > france

4. **label**- The class to which a product belongs. Values belong to the finite set (‘books’,’music’,‘videos’,’rest’)

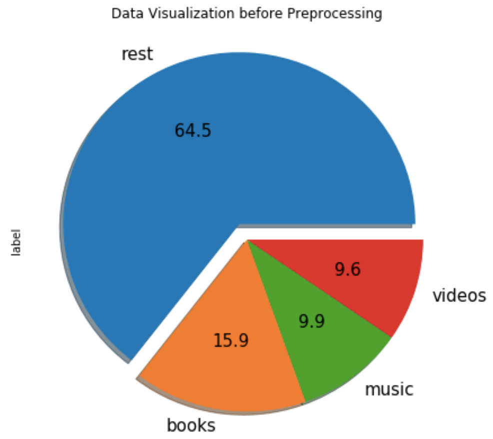
It is possible that for some products only one among (2) or (3) might be available. It means that we may not have data for features 2 and 3 some times.

The problem statement is to classify the products into any one of the buckets
(i) Books
(ii) Music
(iii) Videos
(iv) Rest - A default class for products which doesn’t belong to (i),(ii) or (iii) category.

Exploratory Visualization

- From the below Pie char visualization it is clear that more than 50% of data is un-labelled or it doesn’t have any corresponding label and on the other hand remaining 50% we could see that we have more samples related to books category.

- It is also clear that the dataset is unbalanced.



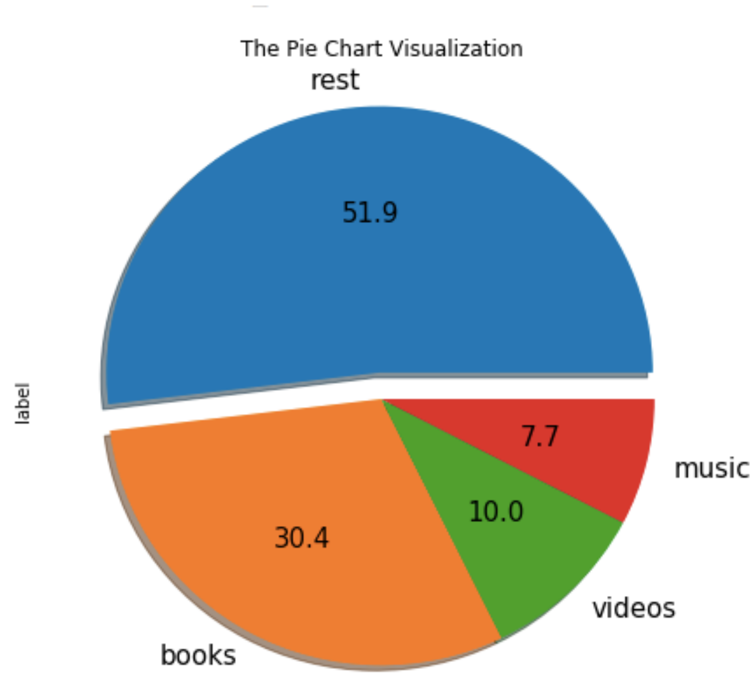
Feature Selection

- From the data exploration, it is straight forward that store id, url and additional attribute features (most of these features are NaNs) don't contribute to predict the label of a product.
- The feature breadcrumbs is right feature to choose to determines label of the unknown product as it clearly describes.

Data Pre-processing

- In the data processing we drop all the null or NaN rows of breadcrumb and lable columns, we can ignore other columns they are not contributing to predict a label.
- In the second step we remove all the duplicate rows by considering label and breadcrumb columns as they are our point of interest.
- As the data of breadcrumb feature contains special characters, numbers, it would be wise to remove all such noise from the data, I have taken care of this step during the tokenization.

Training Dataset Visualization after Data pre-processing



```
Original Dataset size:  603201
Dataset size after noise removal:  599800
Dataset size after duplicate removal:  63126
```

Training and Testing Data Split

As it is a rule of thumb to split the original input data set into training and testing data sets, where training dataset size should be 80% of the original data and testing dataset size would be remaining 20%.

```
Training set has 50500 samples.
Testing set has 12626 samples.
```

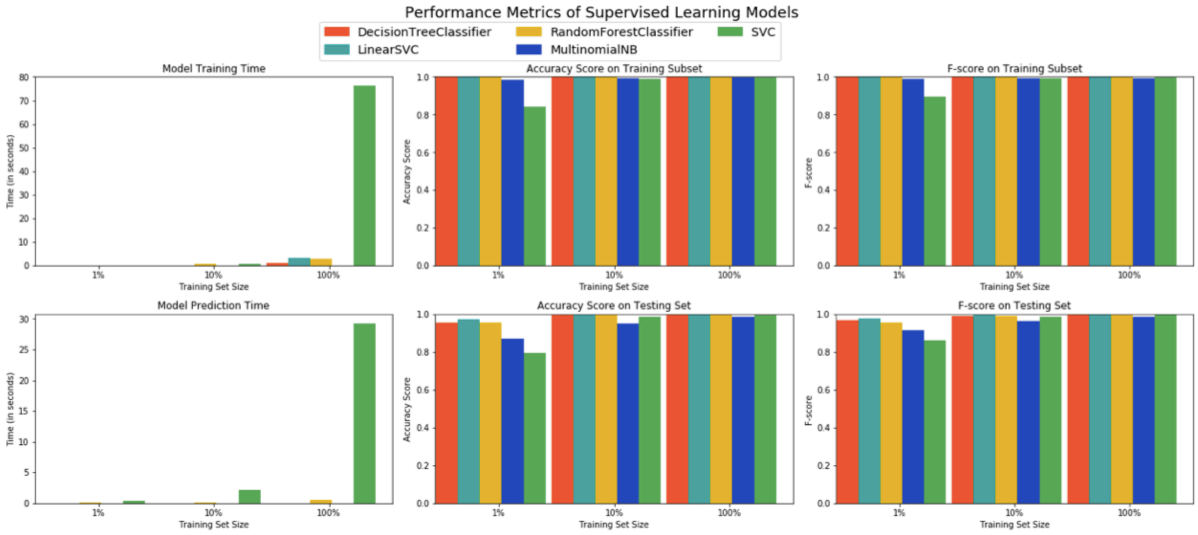
Feature Transformation

- As we are dealing with the text or categorical data it is a rule of thumb to transform all such data into numerical form so we can feed it into the machine learning algorithms, as they work only on numerical inputs.

- We use CountVectorizer of sklearn machine learning library, where it construct a Document Term Matrix and construct Bag of Words for the input data.
- After transforming the breadcrumb data we have to transform the labels of each category to a numerical form, We achieve this by using the Label Encoding module of sklearn.

Model Training

Once we perform data pre-processing and transform, we try to fit or train a couple of classification models and check which is performing good on the given dataset. You could see the performance metrics of various models fit on the 10%, 50% and 100% of dataset.



Model Evaluation

- I would like to use accuracy and f-beta score as an evaluation metrics to assess the performance of the classifier.
- Accuracy is a common metric for binary classifiers; it takes into account both true positives and true negatives with equal weight.
- accuracy= true positives + true negatives/dataset size.
- As the dataset is unbalanced I would like to use f-beta score as the another evaluation metric, with beta being 0.5.

Model Selection and Hyper Parameter Tuning

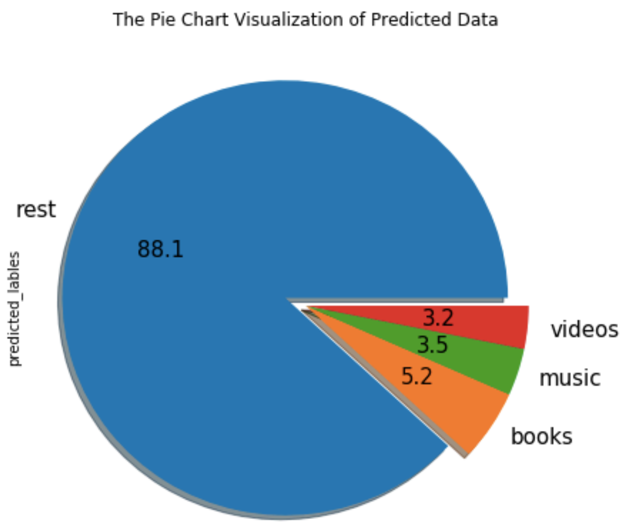
- By taking the models accuracy and f-beta score into consideration, I have selected Decision Tree Classifier is the right fit for my problem.
- So, by using gird search I have performed the hyper parameter tuning to get the best parameter combination of the model.
- It is noticed that even after taking the best parameter combination returned by grid search, the accuracy and f-beta scores of the optimized model is not improved, however by using Decision Tree Classifier I could achieve 99.87% accuracy and 0.9987 f-beta score.

```
Un-optimized model
------
Accuracy score on testing data: 0.9987
F-score on testing data: 0.9987

Optimized Model
------
Final accuracy score on the testing data: 0.9987
Final F-score on the testing data: 0.9987
gird search completed
```

Prediction

- Perform the prediction and lable the products with the most relevant label appropriately.


The Pie Chart Visualization of Predicted Data
predicted_lables
rest
videos
music
books