Vector Image Subject Classification and Popularity

ML 2020: R/W Supervised

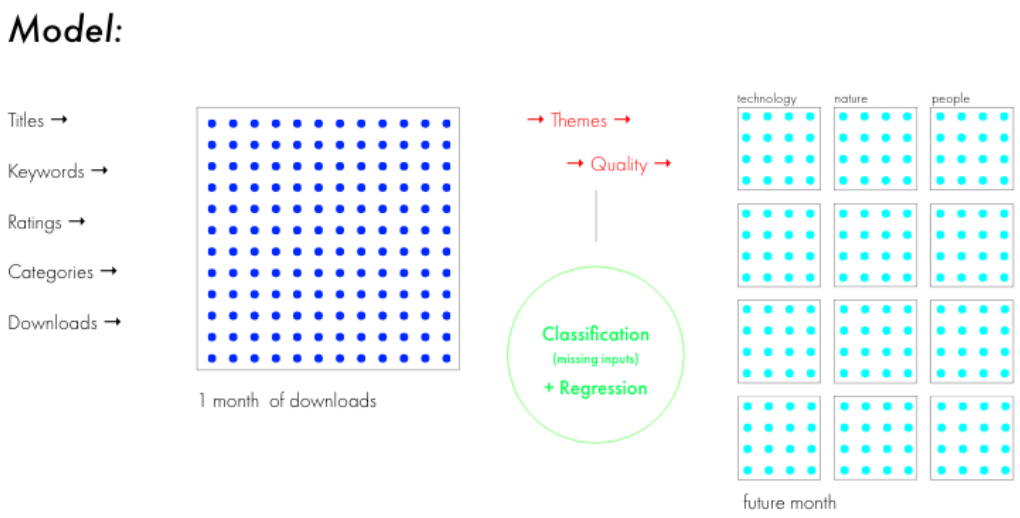
Shea Molloy

A real-life application of machine learning could very well be found in my day job at Adobe, where I work on research and strategy for vectors and illustrations for Adobe Stock. Content creators upload images and editable files to Adobe Stock to be sold in content marketplace crafted for creative professionals. These images are processed through our moderation team at a massive scale: approximately 1.15 million images are uploaded per month, with 1.6 images downloaded per month. Last year's monthly averages were ~800k images uploaded per month and ~900k images downloaded per month. In 2020, there has so far been an average of ~1.2m images uploaded and ~1.3m images downloaded.

Each image has category and subject metadata: keywords and titles written by creators, and a quality rating assigned by moderators. Quality is rated on a scale of 1-5 by the moderation team and is determined by a review of aesthetic and technical factors. With submission and download rates increasing, we should optimize operational benefit to offset the cost (employee time). As we push for salaried compensation for the moderation team, we can glean useful insight from the ingestion data they contribute to the strategy, operations, and business development teams. Knowing that these image collections are on the rise and that there is a fair amount of informative metadata associated with each image, I would like to predict which themes are popular each month to encourage relevant high-quality uploads within those themes.

In order to accomplish this, I would first assess a previous month's popularly downloaded themes and growth. By defining theme buckets using the **bag-of-words** text classification model, beginning by using CountVectorizer to collect all the nouns in the titles of images downloaded, we can define a feature as title_nouns. After collecting all the nouns, I would use VotingClassifier to define useful theme buckets with the frequency in which themes occur and Label Propagation to map keywords to these buckets. We can refer to these features as theme_buckets and theme_details, respectively.

"Categories" are already sometimes defined in the individual vector assets via dropdown input by the moderation team, though they are often missing from the ingestion process. We can use a regression model to predict the categories that are missing based on the categories identified some of the vectors. By ingesting categories and their related metadata, we can add the missing categories and incorporate them into our new, more specific theme groupings, a feature we will refer to as **category_input**. Grouping the nouns from the titles of the assets, and mapping those nouns to the metadatas and applying the categories sometimes missing from individual assets will give us a clearer picture of what themes in a more relevant structure to buyers' needs than just looking at patterns in categories defined by developers and designers at Adobe.



To make these themes more meaningful from a creative direction perspective, we would weight the downloads of images labeled "4", "5", or "Premium" as more important in defining themes than those labeled "1", "2", or "3" using weighted average probabilities within another **VotingClassifier** algorithm. We will call this feature the **quality_weight**. Low-rated downloads would be made more important to defining themes in relation to the quantity of their downloads using a **SGDClassifier**. We will refer to this feature as a **usefulness_weight**. Between high-quality content downloaded at normal rates and low-quality content downloaded at high rates, we will be able to identify what content will be most relevant for the given month.

Knowing the relevant themes from a previous year, we would look at year-to-year growth and apply a **linear regression** model to the current year's month. For example, if high-quality images associated with "health" accounted for 10% of the 100k images downloaded in March of 2019 and we are expecting the collection to grow by 15% in March 2020, we would want to make sure the collection supported 15,000 high-quality health downloads. The growth itself would be predicted by the past 12 months' increase in downloads, a feature we can then call **predicted_growth**. By taking into account the rate of the increase in downloads, we will be able to examine themes that have potential for more growth due to their relation to other fast-growing themes. We can add this as a weight in our prediction using and refer to this feature as theme_relevance.

By combining the classification models to define themes and growth prediction of a regression model, we gain new insight into what content will be most relevant for Adobe Stock buyers. Generating themes by filling in missing data, sifting through titles and mapping keywords is infinitely more relevant than making predictions based on spotty data alone, and taking the growth into account makes this model more accurate to current content patterns than Adobe Stock metadata inputs currently allow. I would hope that a project of this complexity would lend itself well to content intelligence projections that would improve the strategy of the Adobe Stock content team as a whole.