iAuditor

by SafetyCulture

Data Scientist Challenge

Template Data Extraction:

I have started my analysis by getting important set of features from given JSON file. Using the given data description, I found that the data is in hierarchy, hence; I decided to convert the file in appropriate format which can be further processed, and insights can be derived. By looking at the JSON file, I found that most of the data is not useful for text processing. JSON data can have missing fields, thus, before extracting fields, I did analysis on missing keys from JSON file. I found that only some of the keys exists in all templates and I decided to choose subset from those only. Selected subset of template fields shown below.

Extracted template information

- template_id
- template_data
    - metadata
        - name
        - subindustry
        - industry
    - metrics
        - use_count
        - rating
- items
    - label

I have not selected rest of the fields due to any of the following reasons

- date-time information
- unique identifiers
- meta data
- blank values

From the selected subset of fields, I have done text processing on 'label' field under 'items' hierarchy. I found 'label' field useful as it contains checklist text. I haven't selected all the 'items' except which has type 'question' in it. The reason behind doing that other types are not useful for further analysis as they contained information and instructions for end users. In addition, I have used google stop words list (https://code.google.com/archive/p/stop-words/) to remove all unnecessary words from 'label' field. Then, I stored these fields in separate text file which can be used later for further analysis.

After deriving these fields, I opened csv file and tried analyzing using by industry and sub-industry. I did this step to find out whether a dataset is labeled or not. Initially, I thought I can use industry and sub-industry as a label, but that was not the case as templates under same industry and sub-industry were diverse. Moreover, using industry and sub-industry as label for text classification would not be an ideal choice according to my opinion hence, I decided to use text clustering.

Data pre-processing:

1. The text contains numeric values, hence; I used regular expression to remove symbols and numbers which are not useful in text processing.
2. I used lemmatization technique to find root words for different words which have the same origin. The purpose behind using lemmatization is it can be useful in vectorization methods where we are required to convert our text to vector by using word2vec, TfIdf, or GloveVectorizer.
3. I have used Tf-Idf vectorization technique to convert the text into sparse matrix. Tf-Idf is really useful and incorporate the term importance in a document as well as term occurrence over the dataset.
4. After applying Tf-Idf, I removed all the templates which didn't have any occurrence of words.
5. In further processing, I applied single value decomposition technique to data matrix. This technique has ability to reduce the dimensionality of matrix so that we can reduce our matrix features to specific size. I tried different number of components to analyse the performance impact and I found 100 as a suitable choice.

Text Clustering

After getting matrix from single value decomposition, I decided to apply clustering to cluster similar templates for recommendations. KMeans clustering is a good starting point to create clusters and uses Euclidean distance to generate clusters. As our data is a numerical sparse matrix, KMeans is one of the suitable choices. The idea is getting cluster number for every template and recommending similar types of templates to user based on user's previous history. For new users, we can recommend templates from different clusters to give them diversity and choices.

In order to get optimal number of clusters, I used well-known elbow method. I selected a range for number of clusters to derive cluster inertia and silhouette score. Inertia is useful as it gets sum of squared distance from nearest clusters and Silhouette score sense of overlapping clusters. By plotting, these two measures, I derived number of clusters. The optimal number of clusters is 9.

Then, I assigned every template to a cluster and stored in clustered_templates_data.csv file.

Questions and Answers

1. We're interested in seeking solutions that help increase the number of customers using the standard public checklist templates for their inspections (e.g. how can we recommend the suitable templates to customers)?

Ans – We can use the above-described method to find similar clusters of templates and recommend them to users depending on the templates they have used. For new users, we can offer them a diverse set of templates to get started and once we have the template history, based on that, we can suggest the templates from a similar cluster. Another approach is to use collaborative filtering. Using user-user collaborative filtering, we can find templates used by similar categories of users based on user characteristics and recommend it to users with the same characteristics.

2. Based on the findings above, do you have any suggestions on potential features/products that can be built to improve our customers' experiences?

Ans:

We can integrate a feature into the application that will recommend the checklist items to users who are creating custom checklist items based on their industry and sub-industry. If we don't have information about the industry or sub-industry, we can use sequence modeling techniques to train the model on templates data. The benefit of doing this is we can recommend the user next word or sequence of words based on the custom checklist he\she has filled. By doing these, we can help customers to fill their custom checklists and improve the user experience by making a process much faster.