Industry Classification

Overview:

Job title Classification, in this task I used SGD and MLP models to classify job titles and then made a restful API service where input is http request with jobtitle parameter return the industry.

Techniques for cleaning the data:

I cleaned the data using regex and simple string processing in python in the following steps:

1 – First, I removed the leading and trailing space.

2 – I noticed that probably after ‘-‘ nothing is important to the job title, so I removed everything after ‘-‘

3 – Converted all string to lower case to make sure all in lower case.

4 – made a list of words that aren’t important, like some stop words in English and senior or junior, at the end added manager to the list, because it was confusing the classifier, example: sales manager was in 3 categories, most of it was in marketing but because manager has more weight and mentioned in IT a lot, the classifier took every manager as it. Which is not true.

5 – By using re library removed all special characters.

Note: removed duplication only from train and validation.

Note2: the task didn’t seem to need nltk to me.

Classifier:

I didn’t choose a specific classifier at first, tried 2 different classifiers with the default parameters.

1 – MLP:

As known NLP depend a lot and gives very good results with neural networks, so I tried this at first with pipeline having the SMOTE function to handle imbalance. It gave me reasonable results without tuning. The problem was with grid search and to tune this classifier, it kept running for hours without any output. So at the end I tried to tune it manually with different approaches.

2 – SGD:

My second try was SGD Classifier because it contains many classifiers and easy to gridsearch (tune) it also gave me good results without tuning. So I used it with grid search and class weight parameter as balanced to overcome imbalance.

Dealing with imbalance learning:

I try the possible approaches, at first with the MLP I used oversampling minority with the SMOTE function.

With SGD, I used the class weights to be balanced inside the model itself.

Personally I prefer class weight approach because SMOTE have problems but in my works I always try both to make sure.

In the task it seemed like it didn’t matter, both did well with the two classifiers.

How to make model with better performance?

1 – Work more on data cleaning (adding more features) to overcome confusion example: between sales engineer and sales in general, also there is sales in education.

2 – Add more data, without duplication.

3 – Trying more algorithms and tuning all of them.

4 – Cross validation.

Model evaluation:

Accuracy isn’t the best evaluation metric for imbalanced data, so at first I focused on recall as it is giving me how good my model at positive prediction. After the precision dropped I had to go with tradeoff between both, using the harmonic mean f1 score.

Problems and limitations:

1 – Needed more features, as many categories had the same title, so needed more information with the job title which confused the classifiers.

2 – the data is biased towards IT, also the problem if using stop words in English may mistake IT in title for ‘it’ and remove it!

3 – Not so sure about this, but maybe the data is somewhat wrong, not all sales engineers belong to IT industry, so the data needed more target classes.

Notes: removed the MLP model as joblib because the size.