iNetworks

Job title Classification by industry Report

Ahmed Magdy Ali

September 20, 2021

Problem Definition

Given a dataset consists of different job titles, we need to classify each job title to one of four industries which are:

- IT
- Marketing
- Education
- Accountancy

Insights

From the problem definition and after checking the dataset I concluded that this is a supervised learning classification NLP task as the dataset contains the target label, also the features are in the form of text data.

Solution Steps:

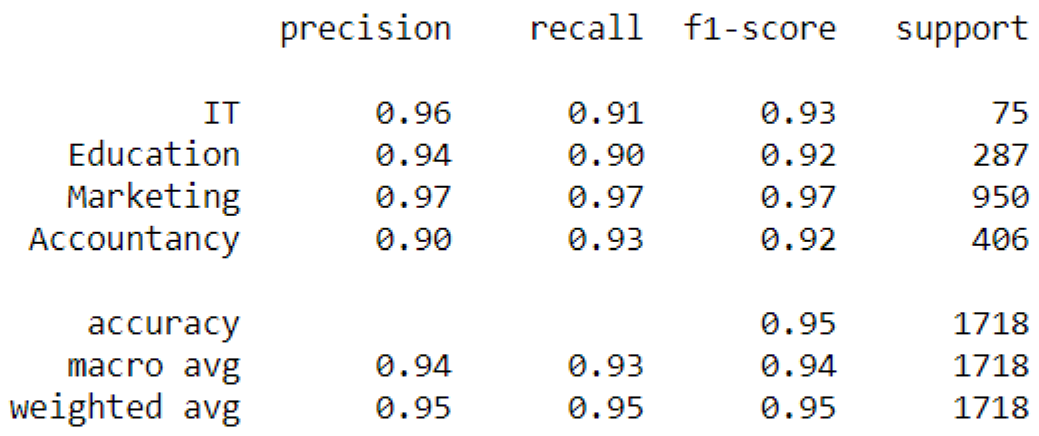
- Data Exploration:

  I checked the data carefully and found that it consists of 8586 entries with no missing data, also there are four unique values in the target label, after further exploration I found that the data is unbalanced among the four classes, also I found that the job title column has a lot of punctuation which is inappropriate and meaningless to the model, also I checked for upper case letters and found that all the letters were small case.

- Feature Engineering and Preprocessing:
    - First, I decided to deal with the punctuation problem by using regular expression to replace each punctuation symbol with a space.
    - Second, I had to tokenize data into words using nltk package word tokenize to split the sentences in the job title column into separate words.
    - Third, I removed the stop word like am, I, is, etc.... from the words returned from word tokenizer as these words are meaningless for the model.
    - Fourth, I used WordNet_Lemmatizer from the nltk package to stem the words to return all the different variations of a word to its origin so that the model deals with them as the same word.
    - Finally, I split the data into train and test into 80% train and 20% test using the stratified method to keep the balance of classes as it is, then used SMOTEN package to balance the dataset a little bit by oversampling accountancy, education and marketing classes on the training set only after splitting to avoid target leakage and make sure that my model generalizes for any data.

- Model and evaluation:
  - I created a pipeline with 2 steps, the first step is vectorizer which I used TfidfVectorizer to transform the text data to td-idf matrix to be able to use it with machine learning models and I specified the ngram range to (1,3) ngrams so that the vectorizer takes the combination of 1 to 3 words into account to have more meaningful words.
  - At the second step I specified Linear SVM classifier with regularization parameter C of 4 to avoid overfitting.
  - I used cross validation score which uses accuracy metric to evaluate the model with 5 cross validations and got scores:

Validation accuracy	94.5%
Test accuracy	95%

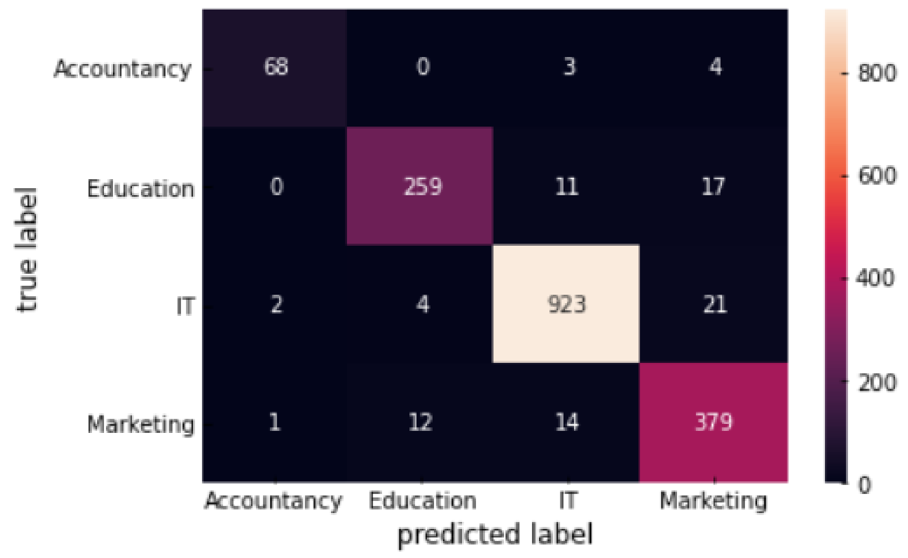


```
              precision    recall  f1-score   support

          IT       0.96      0.91      0.93        75
   Education       0.94      0.90      0.92       287
   Marketing       0.97      0.97      0.97       950
 Accountancy       0.90      0.93      0.92       406

    accuracy                           0.95      1718
   macro avg       0.94      0.93      0.94      1718
weighted avg       0.95      0.95      0.95      1718
```

Confusion matrix:

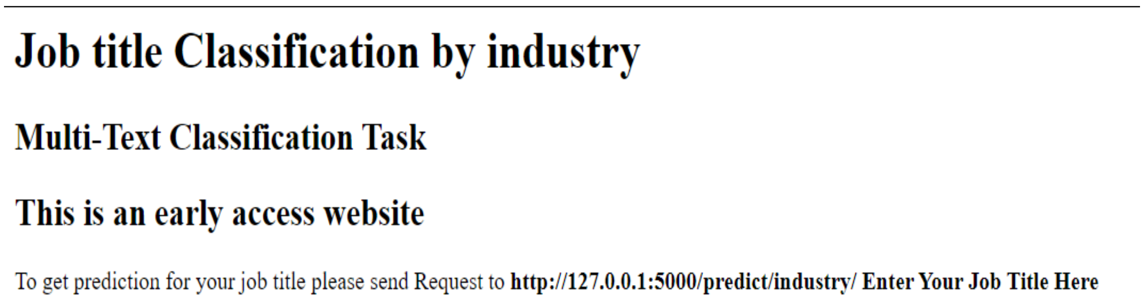


The questions to answer

- Which techniques you have used while cleaning the data if you have cleaned it?
  - Remove punctuation, tokenization, stop words removing, lemmitizer.
- Why have you chosen this classifier?
  - I used Linear SVM Classifier as often text data are linearly separable, and it is a fast classifier and has few parameters to tune, also it performed well on both validation and test set.
- How do you deal with (Imbalance learning)?
  - I used SMOTEN library to oversample the classes with has less share in data compared to other to balance the dataset

- How can you extend the model to have better performance?
    - Try to collect more data to balance the dataset naturally and train more, also I may try to do better preprocessing techniques on the data like trying different kind of balancing and tokenization techniques.
- How do you evaluate your model?
    - It is depends on the case, here I think the best evaluation metric is accuracy as iam concerned about how many right predictions I make from the whole test set, specially that I have balanced the training data and the after plotting the confusion matrix I found that the majority class and the majority of wrong predictions are on the same class which is IT class so the accuracy metric is suitable for such case.
- What are the limitations of your methodology or Where does your approach fail?
    - I used oversampling on some classes so I decided to create artificial data to balance my dataset and avoiding to have a biased model towards one class than the others

Some Screenshots of the model results

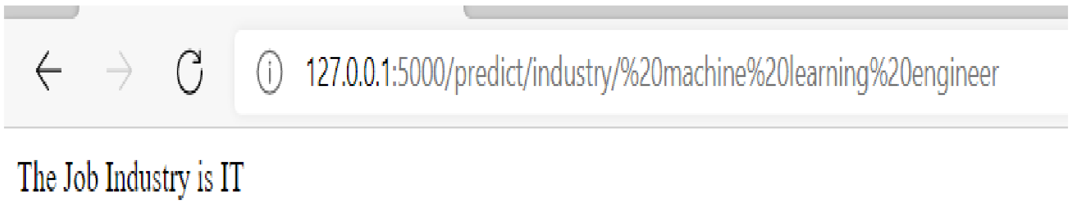


Job title Classification by industry

Multi-Text Classification Task

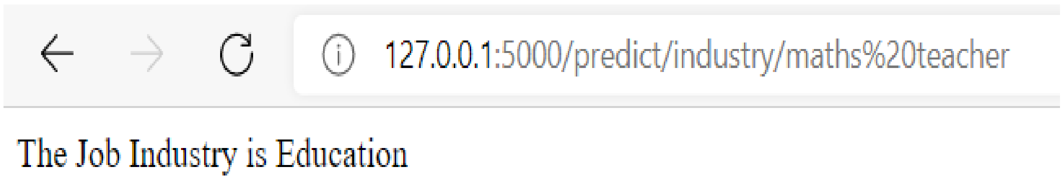
This is an early access website

To get prediction for your job title please send Request to **http://127.0.0.1:5000/predict/industry/ Enter Your Job Title Here**



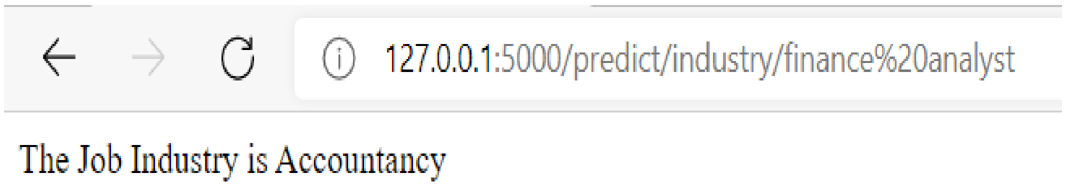
127.0.0.1:5000/predict/industry/%20machine%20learning%20engineer

The Job Industry is IT



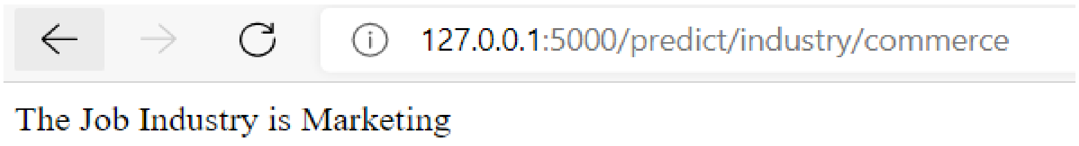
127.0.0.1:5000/predict/industry/maths%20teacher

The Job Industry is Education



127.0.0.1:5000/predict/industry/finance%20analyst

The Job Industry is Accountancy



127.0.0.1:5000/predict/industry/commerce

The Job Industry is Marketing