Text Classification Task

Working Industry Prediction From Job Title

# Questions Answers:

* **Which techniques you have used while cleaning the data if you have cleaned it?**
  + **I used regex over the regular methods as it’s more powerful than the other techniques to clean “job title” column from undesired words/characters mainly to remove the salaries, I also decided to keep company names taking in concertation I only have around 8000 words in the dataset also it may contain words related to the industry which will make the model even better,** **I also notes that we have same job title for different categories EG. IOS developer wrongly labeled as Marketing, I fixed this job titles to fall only in one category, some jobs belong to classes doesn’t exist in the data I thought dropping them will be the best way to handle them.**
* **Why have you chosen this classifier?**
  + **At first, I thought about 4 that would fit this problem, which are Support Vector Machine, Multinomial Naïve Bayes, Logistic Regression and Deep Neural Network, but I excluded the Naïve Bayes because this classifier assumes the words in the text are independent and has no relation between each other which is not the case in the data, also excluded the Logistic regression classifier as it’s doesn’t contain hyper-parameter making it less flexible in case of imbalanced data, while excluding DNN due to the size of the data, I found that support vector machine will be good model for our case due to:**
    - **1. it doesn’t assume that each word is independent meaning it tries to find the relations between the words in the same line**
    - **2. Contain only two hyper parameter “C and Gamma” makes it easer to tune also flexible with imbalanced data**
* **How do you deal with (Imbalance learning)?**
  + **Imbalance learning can be treated by many ways but, I used class weight to give each class appropriate weight, I also used text data augmentation by replacing 2 randomly selected words by its Synonym done this only for the minor class to because I already have enough word in the 3 major classes, text augmentation greatly improved the precision, recall and f1 scores for the minor class, I also notes the existence of duplicates in the data within the same category but decided to keep them in our case as** the purpose of the training set is to accumulate real-life experience, which we will not achieve if you lose the frequency information.
* **How can you extend the model to have better performance?**
  + **The Marketing and the IT classes suffer from over lapping in the job title or to be specific lots of the words used to describe IT jobs has similar jobs containing the same word so traying to increase the number of words describing these jobs (translate the job title to different language then translate it back may help) may improve the performance, also if we can get enough data, we can use RNN or LSTM deep learning models, better tuning for the hyper parameter**
* **How do you evaluate your model?**
  + **I evaluated my model through confusion matrix, precision, recall, f1 and accuracy scores, can’t relay only on accuracy when handling imbalanced data so measuring precision, recall and f1 score gives an indication and help me stop the model from predicting the major class to any input, confusion matrix helps in detecting over lapping between classes.**
* **What are the limitations of your methodology or where does your approach fail?**
  + **My methodology:**
    - **1. can’t handle overlapping between classes (noise data) probably and this is already happen between Marketing and IT classes**
    - **2. not suitable for large datasets as SVM not suited for large data**
    - **3. Can also fail if the text has more words than the number of samples**
    - **4. Don’t have general class so jobs like receptionist, and manager assistant get classified wrongly**