03-anlp20-worksheet

November 18, 2020

1 Work Sheet for Week 03: Text Classification & Sentiment Analysis + Logistic Regression

1.1 Background

In this worksheet, we will follow the material covered either in Chapter 4 of the JM book or in the Text Classification and Sentiment Analysis Playlist. For the questions about logistic regression, you can refer to Chapter 5 of the JM book.

Please keep the provided files in your worksheet folder to be able to run the given scripts and see the text examples without any errors.

Exercises

[E1] You may have noticed that our classification models require a significant number of training examples in order to converge to the underlying probability distribution in the dataset. This brings about the necessity and importance of annotated data: Models require more and more annotated examples each day.

In this exercise, you'll have the chance to annotate your own small dataset (which is basically taken from this Kaggle competition and represented as a json file that includes only 30 examples). Please answer the questions [E1a] and [E1b], after loading the dataset by using the following script:

[3]:
```
import json
import pandas as pd

pd.set_option('display.max_colwidth', 100)

# open JSON file
f = open('movie_reviews.json',)

# converts JSON object to dictionary,  and then to a dataframe

reviews = json.load(f)

movie_reviews_df = pd.DataFrame(reviews['movie_reviews'])

movie_reviews_df.style.hide_index()
```

[3]: <pandas.io.formats.style.Styler at 0x2472c47c548>

[E1a] Try to annotate each movie review example in the dataframe by only using two class labels: 'Positive' or 'Negative'. Write down your annotations as a list of pairs of ID of review, numerical encoding of label, so that (1,0) might mean "example 1 is annotated with label 0. You can select 0 as the negative and 1 as the positive label for convenience. Then compare it with the results of your peers. Can you think of a way of quantifying similarities and differences between your annotations?

[ ]:

[E1b] Imagine doing the task at [E1a] once more, this time having five class labels: 'Positive', 'Somewhat Positive', 'Neutral', 'Somewhat Negative' or 'Negative'. How would you think the differences with respect to your colleagues change? How would you set guidelines for a more robust annotation process, considering also certain arbitration rules for examples with seemingly a lot of different annotation candidates?

[ ]:

[E2] In this exercise, you'll be provided with 4 different Turkish newspaper texts: These news belong to 4 different class labels: 'technology', 'sport', 'art and entertainment' and 'economy'.

Assuming that none of you speak Turkish, how would you assign a unique label to each of these texts given in the dataframe below? Run the script to see the dataframe and think about the appropriate label for each text.

[5]:
```
import json
import pandas as pd

pd.set_option('display.max_colwidth', 500)

# open JSON file
f = open('turkish_news_examples.json',)

# converts JSON object to dictionary,  and then to a dataframe

news = json.load(f)

turkish_news_df = pd.DataFrame(news['turkish_news'])

turkish_news_df.at[0, 'label'] = ""
turkish_news_df.at[1, 'label'] = ""
turkish_news_df.at[2, 'label'] = ""
turkish_news_df.at[3, 'label'] = ""


turkish_news_df.style.hide_index()
```

[5]: <pandas.io.formats.style.Styler at 0x2472c4ce188>

How do you make your predictions? Do you look at any particular words to decide on the actual label? How does the way that you solved this task relate to how Naive Bayes does classification?

[ ]:

[E3] If the document is represented by a certain type of feature, there is a direct relationship between a Naive Bayes classifier and class conditional unigram models. How? Did you use such a relationship while solving [E2]?

In other words, you need to consider a given text document represented by a certain type of feature (e.g. words themselves, parts of speech of words, pronunciations etc.), and our text classification model works with the mere counts of each type in the feature vocabulary. Would such a model resemble to what we have experienced so far with unigram models?

[ ]:

[E4] What's the problem with correlated features for NB, and how may those occur in natural language? Hint: If you are not sure about how correlated features effect the performance of NB classifiers, you can check out this link to get some ideas about how to design an NB algorithm.

[ ]:

[E5] Derive the gradient for binary logistic regression.

[ ]:

[E6] Derive the gradient for multi-class logistic regression.

[ ]: