Course Project

Project description

The Film Junky Union, a new edgy community for classic movie enthusiasts, is developing a system for filtering and categorizing movie reviews. The goal is to train a model to automatically detect negative reviews. You'll be using a dataset of IMBD movie reviews with polarity labelling to build a model for classifying positive and negative reviews. It will need to reach an F1 score of at least 0.85.

Project Instructions

1. Load the data.
2. Preprocess the data, if required.
3. Conduct an EDA and make your conclusion on the class imbalance.
4. Preprocess the data for modeling.
5. Train at least three different models for the given train dataset.
6. Test the models for the given test dataset.
7. Compose a few of your own reviews and classify them with all the models.
8. Check for differences between the testing results of models in the above two points. Try to explain them.
9. Present your findings.

Important! The project template already contains some code snippets for your convenience, so you can use them if you'd like. If you want to start right away from a clean sheet, just remove all those code snippets. Here's the list of code snippets:

- A bit of EDA with some plots

- `evaluate_model()` - a routine to evaluate a classification model which conforms to the scikit-learn predict interface
- `BERT_text_to_embeddings()` - a routing to convert a list of texts into embedding with BERT

The main task to build and evaluate models is your job.

As you can see from the project template, we suggest trying classification models based on logistic regression and gradient boosting, but feel free to try other methods. You can mess around with the project template's structure as long as the project's instructions are completed.

You don't have to use BERT for the project because it is very demanding for computational power and will be very slow on the CPU for the complete dataset. Because of this, BERT usually needs to be executed on GPU for adequate performance. However, you are more than welcome to try and include BERT in the project for a part of the dataset. If you want to do this, we suggest doing so locally and only taking a couple hundred of objects per each part of the (train/test) dataset to avoid waiting too long. Make sure to indicate your use of BERT in the first cell (the header of your project).

Data Description

The data is stored in the `imdb_reviews.tsv` file. Download the dataset.

The data was provided by Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). ***Learning Word Vectors for Sentiment Analysis.*** *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*

Here's the description of the selected fields:

- `review`: the review text
- `pos`: the target, '0' for negative and '1' for positive
- `ds_part`: 'train'/'test' for the train/test part of dataset, correspondingly

There are other fields in the dataset. Feel free to explore them if you'd like.

Project Evaluation

We’ve put together the evaluation criteria for the project. Read them carefully before moving on to the task:

- The text data has been loaded and pre-processed for vectorization.
- The text data has been transformed to vectors.
- The models have been defined, trained, and tested.
- The metric's threshold has been reached.
- All the code cells are arranged in the order of their execution.
- All the code cells can be executed without errors.
- You’ve made conclusions.

Our reviewers will also look at the overall quality of your project:

- Did you stick to the project structure?
- Did you keep your code neat?
- Have you managed to avoid code duplication?
- What are your findings?

You have your takeaway sheets and chapter summaries, so you're ready to proceed to the project.

Good luck!