Implementation of Machine Learning Framework for Record Metadata Classification

Contact Information:

Name : Ankur Shukla
Country : India
Email : work.ankurshukla@gmail.com
Phone : +918291339758
Github : www.github.com/daas-ankur-shukla
Twitter : www.twitter.com/ankurCSRE
LinkedIn : https://www.linkedin.com/in/ankur-shukla-8b73b266/
Skype : daas.shukla

Title:

Implementation of Machine Learning Framework for Record Metadata Classification

Synopsis:

The reason for choosing Zenodo for GSOC is my passion for science and enthusiasm for Open Source. Zenodo is a combination of both science and open source and thus appeals to me a lot. The other factor which motivated me towards Zenodo is my passion for Machine Learning and Python. I have been working with Python for past 6 months and have taken up machine learning as my career objective. I have been learning many concepts of Machine Learning and have also applied some using Python.
My inclination towards Machine Learning motivated me to take up `*Spam Filter and Metadata Classification*` project under Zenodo. I am very enthusiastic about this technical challenge as it involves many dimensions in itself like, data analysis, language processing, implementation of ML concepts and frameworks.
For the GSOC period I propose to build a machine learning framework to integrate the following features in Zenodo:

1. Feature extraction from metadata.
2. Cleaning and Normalization of data

3. Multi-class classification on Zenodo's metadata
4. Storage and retrieval ML models as required
5. Take user input to improve accuracy.

Classifier's test results would be randomly chosen for verification from user, to generate false positive and false negative statistics. The framework will then use these numbers to improve its accuracy.

Benefits to Community:

The proposed framework solves the problem of spam data creeping into Zenodo's database. By classifying a record as spam, we can save on query time, and most importantly can generate a better user experience.
If this framework is implemented, it can serve many purposes in the near future to make Zenodo a more robust service for spam free scientific data, as we can suspect the probability of a record being spam right at the time of its generation and then take actions accordingly.

Deliverables:

In this GSOC I propose to deliver the following:

1. A Machine Learning framework to store, retrieve ML models, perform test queries on a metadata batch, and output classification results
2. Implement a robust multi-class multi-label classifier under the ML framework designed. The classes which the classifier will deal with are following (non-exhaustive):

Class	Label
Spam	Spam, Genuine
Scientific Domain	Biology, Physics, Chemistry
Keywords	Machine Learning, GIS

3. Integration of ML Framework into Zenodo for taking user suggestion for false positives and false negatives. Those records having intermediate probabilities will be displayed to users or administrators. The record label would be modified based on administrator

review. This will also be used to create a confusion matrix which we can use to update model weights.

Timeline:

I have my summer vacations in the month of May and June. In these months I will be able to devote 6-7 hours per day that is around 45 hours per week. After that I will able to devote 4-5 hours in weekdays and around 7 hours in weekends. This makes around 37 hours per week. I will not have any major exams in GSOC period, however I will be having periodic quizzes.

A. Community Bonding Period (May 4-May 29, 2017):
    a. Make a wiki page on the Zenodo repository for fortnightly tasks and report list
    b. Study the Zenodo coding architecture
    c. Study the documentation framework
    d. Discuss the feasibility of different machine learning algorithms for Zenodo's record metadata structure.
    e. Discuss the outline of the framework and its functionalities
    f. Discuss schema for conversion of record metadata to feature matrix for classification algorithm
B. Coding Period (May 30-August 29, 2017):
    a. Coding Period Phase 1 (May 30-June 29, 2017):
        i. Week 1 (May 30-June 5):
            1. Analyse metadata dataset and develop code to extract essential features from dataset
        ii. Week 2 & 3 (June 6-June 19):
            1. Develop code to clean and normalize extracted features
        iii. Week 4 (June 20-June 29):
            1. Test, debug and document code
            2. Prepare work for Phase 1 submission along with a brief Phase 1 report
            3. Prepare Pre-Phase 2 synopsis
    b. Coding Period Phase 2 (June 30-July 27, 2017):
        i. Week 5, 6 & 7 (June 30-July 20):
            1. Discuss the merits and demerits of different machine learning libraries and zero in on one.
            2. Code a multi-class multi-label classifier for metadata classification
            3. Work on optimizing the classifier on various parameters.
            4. Test, and debug the code

ii. Week 8 (July 21-July 27):
    1. Begin implementing a framework on top of the classifier for the following feature (non-exhaustive):
        a. Store and retrieve models
    2. Prepare elaborate documentation and tests for Phase 2 components
    3. Prepare work for Phase 2 submission
    4. Prepare Pre-Phase 3 synopsis

c. Coding Period Phase 3 (July 28-August 29, 2017):
    i. Week 9 & 10 (July 28-August 10):
        1. Finish implementing the framework with the following features (non-exhaustive):
            a. Retrain model by retrieving it and training it with new batch of metadata with known classes
            b. Query test results for batch of record metadata
            c. Output the query result in specified format
        2. Connect the framework to Zenodo service
    ii. Week 11 & 12 (August 11-August 25):
        1. Test, debug code
        2. Write pending documentation
        3. Prepare Final Phase submission along with a detailed final phase report

Related Work:

- To analyze and understand the sample data provided by Zenodo, I am coding a visualizer to (repo):
    - Create feature matrix for the metadata
    - Populate the matrix with entries
    - Select important features from the data
    - Tokenize the important text features using NLTK Python Library
    - Use the tokenized features to create statistical analysis on spam and non-spam data
    - Create visualization to understand the data
- I am also working on the issue #937 for Zenodo. However I have not been able to make a PR uptill now due to my course work. I will soon be making one.

- I have implemented a linear regression model from scratch using numpy with Principle Component Analysis to reduce the data dimensionality, as a part of Machine Learning course (the code is present on my github repo).
- I am currently involved in developing Neural Networks from scratch using numpy for classification on a sample dataset and a deep learning model using Convolutional Neural Networks to make facial expression classifier as a part of the same course.

Biographical Information:

1. **Education:**
    a. **Postgraduate (2016-Present):** M. Tech. in Geoinformatics and Natural Resources Engineering from IIT Bombay
    b. **Undergraduate (2010-2014):** B. Tech. in Electrical Engineering from KNIT Sultanpur
2. **Work Experience (2014-2015):** Assistant System Engineer at Tata Consultancy Services Ltd
    a. Trained in Core and Advanced Java
    b. Developed a tool in Java to check for anomalous data in a Hive database
3. **Open Source Contribution:** My Github Repo - daas-ankur-shukla
    a. Developed a web app using Django, as a smart parking solution based on IoT and PgRouting and presented it in FOSS4G ASIA 2017 conference at IIIT Hyderabad. The same work as been chosen for an international publication by the conference community.
    b. Developed a MATLAB GUI to implement low pass filters on an image and extract low frequency component of an image to achieve smoothing effect.
    c. Helped convert a C++ code to STL for 2.4 release of PgRouting (PR)
4. **Coding Skills:**
    a. Python:
        i. Image Processing Libraries such as GDAL
        ii. Django, Flask
    b. MATLAB:
        i. Simulink
        ii. GUIDE
    c. Java:

    i. Core and Advanced concepts
    ii. Struts, Hibernate
d. Javascript:
    i. OpenLayers 3.0