CS 410 Project Proposal: Leaderboard Competition for Detecting Sarcasm in News Headlines and Articles

Greg Lee (gglee2) [Captain]

1 Problem Definition

My proposed project is a Leaderboard Competition hosted on LiveDataLab. The task that participants are solving is training a model which predicts whether a new article is sarcastic based on its headline and contents. In other words, this is a binary classification task: given a news article, classify it as "Sarcastic" or "Not Sarcastic". This is an important problem because many people consume news at a glance and may unintentionally spread misinformation if they don't realize an article is satirical. Additionally, for web scraping tools that aggregate news, it would be extremely useful to have a tool which can automatically detect (and filter out) articles which are not intended to be serious news.

2 Dataset and Participant Interaction

The main inspiration and source of data for this project comes from this Kaggle Dataset. This data is publicly available through Kaggle but the author requests these two articles be cited if the data is used: Sarcasm Detection using Hybrid Neural Network [1] and Sculpting Data for ML [2]. The dataset provides tagged news articles sourced from a satirical news website, The Onion, and a non-sarcastic website, the Huffington Post. An issue present for collecting data for this task is that tagging news articles requires manual human supervision to ensure accurate tags for training data. By sourcing the data from a website dedicated to sarcastic articles and one dedicated to "real" news, we have very reliable class labels to work with. This is susceptible to potentially overfitting to these two sources, so future work could include expanding the scope of this data.

Each record in the dataset is represented by a JSON object with three attributes: is_sarcastic, headline, and article_link. is_sarcastic is the 0 or 1 binary class label where 1 denotes a sarcastic article. The headline is a string for the headline of the article and article_link is the url of the original news article. Participants are expected to take the headline and article link as input, and output a class label representing sarcasm. They can use techniques from the course

to analyze the text found in the headline and the article contents (scrape the URL with a tool like BeautifulSoup). Participants will clone a simple template GitHub repository and add a LiveDataLab webhook (just like in the course MPs). Then they can implement whichever models they like in a Python file named `classifier.py`. On each commit, the webhook will fire: the model will be trained on a training dataset and evaluated on a test dataset using a variety of classification evaluation metrics. Performance on both the training and test data will be presented to the user on the leaderboard.

3 Implementation Plan

The first step to creating this Leaderboard Competition is to perform exploratory data analysis on the news dataset and implementing a very simple baseline to debug with. Afterward, I'll implement the binary classification metrics that participants' models will be evaluated with. Namely, the evaluation metrics used will be Accuracy, ROC AUC Score, and F1 Score (for ease of interpretability, all of these metrics take a range of [0, 1]). A submission's placement on the leaderboard is based on the average of these three scores. Overall the columns on the leaderboard would be: Rank, Username, Submission Number, Test Acc, Test ROC AUC, Test F1, and Overall Score.

After this point, the rest of the project would entail integrating these pieces of code with LiveDataLab. After the webhook triggers, a program should run the participant's code to train it on a training set (80%) and then evaluate it on the test set (20%) using the three evaluation metrics (and then averaging those into an Overall Score). Note: I'm not exactly sure how LiveDataLab works on the backend. This is the place where I'm looking for the most feedback from course staff.

Planned Timeline

Task	Due Date
Submit Progress Report/Team Formation	10/22
Exploratory Data Analysis	10/26
Develop Simple Baseline	11/4
Implement Evaluation Methods	11/10
Submit Progress Report	11/14
Integrate Code with LiveDataLab	11/21
Finalize Documentation	11/25
Presentation	11/30
Submit Final Project	12/7

References

[1] Rishabh Misra and Prahal Arora. Sarcasm detection using hybrid neural network. *arXiv preprint arXiv:1908.07414*, 2019.

[2] Rishabh Misra and Jigyasa Grover. *Sculpting Data for ML: The first act of Machine Learning*. 01 2021.