PROJECT PROPOSAL

Domain Background: Natural Language Processing

Natural language processing has been a growing field of research with unstructured text data being available everywhere, such as social media, chats, emails, web pages, and survey responses. It can be difficult and time consuming to extract information from raw text due to its unstructured nature. However, with proper analytical techniques and tools applied, a rich source of information can be extracted from text data.

Among many fields in natural language processing, text classification is seen as one of the most common and frequently applied techniques. The main aim of text classification is to assign labels to raw text based on its content. It allows businesses to automatically analyze unstructured text data, allowing for quick and cost effective decision making. Text classification is being applied in many business cases, where some most used cases are:

- Spam detection in emails
- Sentiment classification
    - ✓ Analyze tweets or comments in social media to determine if customers are talking positive or negative about a brand or a topic.¨
    - ✓ Analyze customer feedbacks to from survey results and reviews to understand their like or dislike for the product.
    - ✓ Profanity and Abuse detecting in social media platform
    - ✓ Analyzing support tickets in assigning tags to common questions for quick customer service.

There are abundant academic research performed for text classification in different domains using machine learning. Sentiment analysis of customer reviews to classify the reviews as positive or negative is one such mostly studied area that helps in understanding the customers' liking or disliking to a product.

With customer reviews from Amazon, in the following paper, authors perform sentiment classification to label reviews as positive or negative using machine learning models Decision tree, naive bayes and support vector machine.

Singla, Zeenia, Sukhchandan Randhawa, and Sushma Jain. "Sentiment analysis of customer product reviews using machine learning." 2017 international conference on intelligent computing and control (I2C2). IEEE, 2017.

Problem Statement

The problem proposed for the capstone project is the 'Toxic Comment Classification' posted on Kaggle. With the growth in use of online platforms, there has also been rise in threat of abuse and harassment online. This results in many people stopping themselves from expressing themselves and give up on seeking different opinions. In addition, online platforms struggle to effectively facilitate the conversations, and are limited or completely shut down user comments.

The main focus of the problem is to identify the level of toxicity in the comments, such as threats, obscenity, insults and identity-based hate. In addition, the challenge is to build multi-headed model, meaning a multi-label task, where a single comment can be classified to multiple labels. The model can help in making online discussion more productive and respectful.

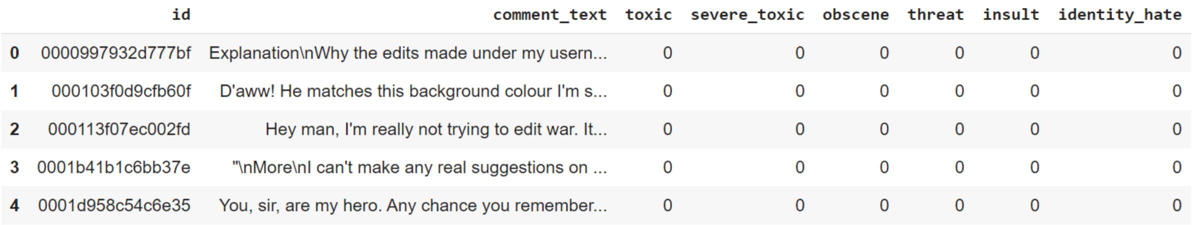
Datasets and Input

Dataset for the given problem is retrieved from Kaggle, which can be downloaded from https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data.

The data contains Wikipedia comments that have been labeled by humans for toxic behavior.

- As an input, the data has raw text, which are the comments.
- The output for the text is given by six different labels of toxicity: Toxic, severe toxic, obscene, threat, insult and identity hate.
- The labels are represented as binary indicators (1, 0).
- The texts (comments) can also have no labels to them, meaning they are clean comments and does not contain any toxic labels.
- Also, since, this is a multi-label classification task, each of the comments can have one or more than one labels.

The image below shows few samples of the raw data.



	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

Solution Statement

As stated above, this is multi-label classification task and also related to natural language processing, so, the main focus would be to apply text classification methods using machine learning. Also, with machine learning, methods applicable to solving multi-label classification task will be applied.

Benchmark model

- To start with the modeling, bag of words method with TFIDF vectorizing approach will be used to process the text.
- As a benchmark model, very commonly used, a simplistic modeling approach, Binary Relevance will be applied with machine learning models, such as Logistic Regression and Naïve Bayes.

Evaluation Metrics

Commonly used evaluation metrics for classification tasks will be used, including

- Accuracy
- AUC score
- Log loss

Project Design

- As with all machine learning projects, the start would be to understanding the data with some exploratory analysis.
    - ➢ Create summary statistics and visualization
    - ➢ Tools: **Pandas, Numpy, Matplotlib, Seaborn**
- After the exploratory analysis, as the input is text data, cleaning and processing the text data would be the next step.
    - ➢ Data cleaning : **nltk, pandas**
    - ➢ Bag of words representation: **sickit-learn, tfidf vectorizer**
    - ➢ Word Embedding: **word2vec, gensim**
- With data being processed, the benchmark model will be applied to get initial results and understand the complexity of the task.
    - ➢ Binary Relevance Modeling: **sickit-multilearn, sickit-learn**
    - ➢ Machine learning models: **Logistic regression, Naïve bayes**
- The next step will be to apply some more advanced modeling methods to get improvements over the benchmark model.
    - ➢ Deep Learning (LSTM): **Tensorflow, keras**
    - ➢ Transfer learning: **BERT model, Tensorflow Hub**
- The results will be compared from all the methods applied to decide on the final solution to the problem. Since, this is problem taken from Kaggle competition, the results would be submitted to Kaggle to receive the results.
- Finally, a simple web application will be created to present the solution to the problem that will take a text as an input and provide the results as probability of the text belonging to the toxic labels.
    - ➢ Use **Streamlit** or **Plotly Dash** for the web app