**Eth. AI, HW3, Misinformation**

**Due Week 8, Thursday, 11:59 PM**

**Goal:** Work with both supervised and unsupervised machine learning models to identify misinformation.

**Resources:** Tutorials on python and sklearn library can be found both in [W3Schools](https://www.w3schools.com/python/default.asp) as well as [here (Python course Slides](https://ind657-my.sharepoint.com/:f:/g/personal/jrusert_pfw_edu/Ei_EBaJU-IJBgpwM2A7mmvUB-GrO9jUSvSq0CgMgwZgwYw?e=DJcAgX)). Also, sklearn has demonstrations of models and functions on the specific model pages as well (see below links).

**Models:** You will be working with 3 types of models for this homework, 2 from the sklearn python library:

[Logistic Regression](https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html)

[KMeans Clustering](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html)

And 1 from the huggingface library:

[Sentence Transformers](https://huggingface.co/sentence-transformers)

This homework consists of 2 parts, but both parts use the same misinformation dataset provided on brightspace. Even though there are 2 parts, there still should only be 1 final report and 1 .py file handed in.

## Part 1: Supervised Learning

**Goal:** The first part is a simple task. Your goal is to train a simple logistic regression model to identify misinformation.

**Coding goals:**

* Use the sklearn library to train a logistic regression model on the misinformation training set (train\_misinfo.csv).
  + You should use the [tfidf-vectorizer](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html) to automatically convert the text to a vector form.
* Produce a score on the training data.
* Test and produce a score on the test dataset (test\_misinfo.csv).

**To be added to the report:**

* Describe the simple process used to train/test the logistic regression models.
* Report the scores for both the training data and the test data in a small table.
* Describe any difficulties faced during the implementation of the model.

## Part 2: Unsupervised Learning

### Part A. Clustering with TF-IDF

**Goal:** We noted (in class) that unsupervised methods such as clustering could be useful to automatically cluster similar examples without explicitly learning what is misinformation and what is not. Your goal is to leverage a clustering algorithm to automatically cluster data and find misinformation.

**Coding goals:**

* Use the sklearn library and Kmeans to cluster the **test dataset** (test\_misinfo.csv) into 2 clusters.
  + You should use tf-idf vectorizer again to represent the data. However, note that you should assume there is **no access to the train dataset.** You should only be relying on the test dataset.
* Once clustered, you should obtain the cluster labels for each example.
  + This can be done using .labels\_ on the kmeans model.
  + Note that these are **cluster** labels and not prediction labels (you do not know these yet).
* Map the cluster labels to the prediction labels. For example, cluster label 0 might map to class label 0 (no misinformation) or might map to class label 1 (misinformation).
  + To do this, assume you have access to a small subsample of 100 examples and their true labels from the test set.
    - You should create this subsample by randomly sampling 50 examples for each label from the test set.
  + Use this subsample to come up with a method that maps the cluster labels to the prediction labels.
* Get the final predictions for the test examples by applying the mapping.
* Obtain the clustering score by comparing the test predictions to the true labels.
* Create a confusion matrix of the predictions versus true labels.

**To be added to the report:**

* Simple description of the steps used to approach the problem. (Do not include code references, English phrases on the approach only.)
  + Description of how you used the subsampled data to map the cluster labels to the class labels.
* The scores of the clustering approach, in a small table with the supervised learning approach.
  + Make observations on which is doing better and add why you believe there is (or isn’t) a difference.
* The confusion matrix for the clustering approach.
  + Make observations on the clustering approach. Is it over predicting misinformation or under-predicting it?

### Part B. Clustering with Sentence Embeddings Representations

**Goal:** A downside of TF-IDF is it uses exact matching for words. Dense embeddings on the other hand do not and might work better for clustering. Your goal is to repeat the clustering process with sentence embedding representations instead of tf-idf representations.

**Coding goals:**

* Repeat the above process (Part A) using [Sentence Transformers](https://huggingface.co/sentence-transformers) instead of tf-idf to represent each example.
  + Sentence transformers leverage transformers to get a contextual embedding representation for a given text. As can be observed from the given example, you simply need to pass texts into the model and it will return the representations.

*# 1. Load a pretrained Sentence Transformer model*  
model = SentenceTransformer("all-MiniLM-L6-v2")  
  
*# The sentences to encode*  
sentences = [  
 "The weather is lovely today.",  
 "It's so sunny outside!",  
 "He drove to the stadium.",  
]  
  
*# 2. Calculate embeddings by calling model.encode()*  
embeddings = model.encode(sentences)

* + You should use the “all-MiniLM-L6-v2” model for the sentence transformers.
  + The rest of the clustering process should be the same.
  + Do note that since it is querying a larger model, this process will take longer than the tf-idf vectorizer.

**To be added to the report:**

* The scores of the sentence embedding clustering approach, in a small table with the supervised learning approach and tf-idf clustering approach.
  + Make observations on which is doing better in terms of clustering and add why you believe there is (or isn’t) a difference.
* The confusion matrix for the sentence embedding clustering approach.
  + Make observations on the clustering approach. Is it over predicting misinformation or under-predicting it?

## To Turn In:

* Python file (**USERNAME\_HW3.py**.) which contains functions related to the above parts. You should include comments and useful function names to differentiate the two parts and clustering subparts.
* Report which contains:
  + The above requirements for each part. Make sure you make it clear (via titles or subheadings) which part you are describing.
  + Discussion of issues or difficulties encountered.
  + A note of any AI tools used in coding or writing.

**Additional Rules (MUST BE FOLLOWED):**

1. All Homeworks should follow the overall [homework guidelines](https://ind657-my.sharepoint.com/:p:/g/personal/jrusert_pfw_edu/EZxf1ZsRXjBEkQLNhEhdTgUBt6U64KiT1DJ1YHtkARgKwA?e=bwHrbf)
2. The code should be written in python 3.
3. If noted, the functions must follow the naming and number of arguments as demonstrated.
4. You should make your code modular to the different steps. (You may have more functions to help your main functions)
5. You should be adding comments to document your code. **If I can’t understand why you perform an action, then I can’t credit you for performing that action.**
6. The report should be readable and reference your code, **without explicitly including code.**
7. You should include your name and homework number in the comments at the beginning of the python file.

**Report**

The reports for the homeworks are necessary to communicate your learning and thinking through of the material. Examples of good reports can be found on brightspace under Additional Resources/Guides. Note that your report style may differ, but it is a good reference to start with.

**Grading**

Assignment will be graded as follows:

|  |  |
| --- | --- |
| **Description** | **Points** |
| **(.py file)** Supervised coding requirements | 5 |
| **(.py file)** Unsupervised coding requirements | 15 |
| **(Report)** Supervised report requirements | 5 |
| **(Report)** Unsupervised report requirements | 15 |
| **(Report)** Other Report Requirements | 5 |
| **(.py file)** Documentation/Code (Comments, functions, etc) | 5 |
| **Total:** | **50** |

* **If the code does not run, it cannot be graded well.** (Many points can be lost if the code cannot be run, as I will not be able to fully test the implementation of the functions).
* **Breaking of the additional rules can result in applied penalties.** (Always make sure you are checking against the rules)

**Suggestions**

* **Documentation is key for showing your effort in this homework.** Make sure you are noting why you make certain decisions all throughout your code.
* The slides for previous classes are posted, so please refer to these and the book for ideas during implementation.
* Start simple, build up complexity. You should always make sure your new ideas being added do not cause your program to crash. So starting simple is the best way to a) maintain the ability to keep your code running, b) add in comments for documentation and thought process as you add more code.
* Work through the homework yourself, rather than sharing ideas (especially not code) with other students. **As a reminder, plagiarism (or sharing) of code is strictly prohibited.** This assignment is complex enough that significant overlap between students will be suspicious.
* If you have not worked with python before, w3schools can help you translate your previous coding experience to python (<https://www.w3schools.com/python/default.asp>)
* Stop by office hours to discuss ideas. I am always happy to help you think through your process!