**Peem’s ADA Project (Last updated: March 21st 2022).**

**Data description**

We have two datasets:

1. 11,434 videos with content, comments, metadata, etc. without labels on understandability.
2. 600 videos with raw labels on PEMAT guideline on understandability, actionability. A separate label for whether a video contains medical information is included.

Both datasets can be linked by URL (i.e. id) of each Youtube video. Here, I used text to denote human-readable, English phrases with semantic meaning (type = string); string to denote non-language. After merging videos appearing multiple times in the dataset, we are left with 8,900 videos.

The dataset consists of the following features:

**Will not be used because of no predictive value:**

1. embeddable (Boolean): Whether a video can be embedded on (i.e. watched directly from) other websites. Only 1% false; otherwise all true.

'trackKind' (text): Type of audio track. 1/3 missing. Of the remaining, 87% are ASR and the rest standard.

1. topicID.
2. 'contentDimension (text) : 2d or 3d. Every video except 1 video is in 2d. This feature has no values.
3. 'contentDuration (text, number): Duration of the video, denoted in PT'X'M(Min)'Y'(Sec)
4. id (string): id of each video. Also noted as video\_id and URL in some of the datasets.
5. isAutoSynced (Boolean):  Auto Sync refers to automatic syncing of audios and visuals. If a video is auto synced in Youtube, then the video uploader syncs both on Youtube. 67% false. 1% true. Otherwise NaN.
6. publicStatsViewable (Boolean): Whether users can view public statistics of video. 88% True and 12% False. ??
7. license (text): Is the video licensed by Youtube or Creative Common? 99% by Youtube.
8. 'contentCaption (Boolean): Whether a video has caption or not.
   1. Every contentCaption has value = 1.

**Content View (content of a video)**

Caption is an explanation for visuals “[sound of wing]”

1. captid (string):
   1. Def’n: ID of Youtube caption.
   2. 36.5% of the videos have no caption id.
2. captsLastUpdated (text, number):
   1. Def’n: The last time a video caption is updated; is stored in the YYYY-MM-DD: Time format.
   2. **NOTE:** If a caption is ever updated, it has a captionID.
3. categoryId (encoding of text):
   1. Def’n: The category each Youtube video belongs to, such as person, blog, science. See Youtube API for more info <https://gist.github.com/dgp/1b24bf2961521bd75d6c>

**\*\*\* NOTE: categoryId differs from relevantTopicId.**

1. 'contentDefinition (string): Consists of two types: hd (high definition) or sd (standard definition). 68% are hd.
2. 'contentLicensed (Boolean)??:
   1. **Def’n:** Whether the content on that channel is licensed. If a video is licensed, it cannot be used for commercial purposes without the permission of the video creator.
3. description (text): Description of videos seen below the rectangle box (e.g., importance of lipid metabolism).
4. keyword (text): Keyword used to search for that video (e.g., diabetes retinopathy)
5. 'publishedAt' (text, number): Date and time a video is published.
6. rank (number): The rank at which the video appears if searched using keyword in incognito mode.
7. 'relevantTopicIds' (encoding of text): Topic ids created by Youtube based on knowledge graph.
8. 'subtitle' (text): The transcript of what is said in the video (e.g., “In this video, we will talk about”). In this dataset, only 700 videos have subtitles.
9. 'title' (text): Title of the video (e.g., “Importance of insulin”).

**Identity**

1. 'video\_id': Same as id.

**Metadata (**

1. 'topicIds' (encoding of text): All except 3 are NaN
2. 'has\_description' (Boolean)\*\*: Whether a video has description.
   1. If the description contains http://, has\_description is automatically set to zero even if it’s followed by an actual text, such as “http://www.handwrittentutorials.com - This video..”
   2. Some videos (around 100) have has\_description = True even though description is empty.
   3. Videos in the labelled set have 15/600 has\_description == True; videos in the unlabeled set have 750/9000 == True.
   4. Among videos with no description, only one is not understandable.
3. 'video\_duration (text, number): Duration of the video
4. 'word\_count' (number) \*\*: How many words appear in the description. This is calculated using Python library.
   1. Cannot handle special symbols.

\*\*\* NOTE: From sentence\_count and word\_count, we can calculate various readability indices. Because these counts are problematic, the following are somewhat unreasonable.

1. 'word\_unique (number)\*\*: Number of unique words in the description of the video.
   1. If the description begins with http://www….., word\_count is automatically set to 0.
2. 'transition\_words (number)\*\*: Number of transition words (think: and, so, but, however, etc.) in the description of the video.
   1. If the description begins with http://www….., word\_count is automatically set to 0.
   2. Many description, though exists, has zero transition words (think: donate here).
3. 'summary\_words (number)\*\*: How many summary words are said.
4. 'active\_verb' (number)\*\*: How many active verbs appear in the description
5. 'sentence\_count' (number) \*\*: How many sentences appear in the description. This is calculated using Python library.
   1. Cannot handle string of words without full stops.
6. 'Kincaid' (number)\*\*: Flesch-Kincaid readability index. Measures the minimum education level required to comprehend the text. The higher Kincaid is, the more difficult a text is to comprehend. See formula using link above.
7. 'FleshReadingEase (number)\*\*:  Flesch reading ease index. Measures how easy it is to read a text. The higher the reading score, the easier a piece of text is to read)
8. 'ARI' (number)\*\*: Automated readability index. The higher ARI is, the more difficult it is to read the text.

**Metadata (channel)**

1. channelCommentCount (number): How many comments a video receives.
   1. The labelled dataset has a British Got Talent video.
   2. Some channels have extremely high viewCount.
2. channelDescription (text): Description of each channel (created by channel owner). Found in About section.
3. channelId (text): ID of the video’s channel.
4. channelPublishedat (text): Time the channel was established.
5. channelSubscriberCount (number): How many subscribers a channel has.
6. channelTitle (text): Name of the channel (e.g., All about Diabetes and Related).
7. channelVideoCount (number): How many videos that video’s channel has posted.
8. channelViewCount (number): How many views a channel posting that particular video have received.
9. 'keyword\_title\_cosine' (Number): Cosine similarity between title and keyword. Doesn’t suffer from out of range problem.
10. 'keyword\_decription\_cosine' (Number): Cosine simiarlity between keyword and description.

**Outcome variables**

1. 'comment\_title\_cosine' (Number):
2. 'comment\_description\_cosine' (Number)
3. 'viewCount (number): How many views the video receives.
4. likeCount (number): How many likes a video receives.
5. dislikeCount (number): How many dislikes a video receives.
6. 'commentCount (text): How many comments have been made on each video.

**Derivative of outcome variables**

1. 'postive\_comment\_count' (Number)
2. 'negative\_comment\_count' (Number)
3. 'comment\_unique\_words' (Number)
4. 'comment\_total\_words' (Number):
5. 'neutral\_comment\_count' (Number):

**Summaries of the descriptive statistics and observations for each variables**

See merge\_and\_cleaned for a description on the PEMAT labels. This is a descriptive statistics on the entire dataset.

**## Useful resource**

1. ARI: https://en.wikipedia.org/wiki/Automated\_readability\_index

2. Flesch reading ease:

3. Kincaid (formally titled: Flesch-Kincaid readability). https://readable.com/readability/flesch-reading-ease-flesch-kincaid-grade-level/

4. Cosine similarity: https://www.machinelearningplus.com/nlp/cosine-similarity/

**Observations**

**## Questions to Nynke and Larry**

**1.** Every derivative of comments is problematic because it contains special symbols, https://, etc. I'm thinking about spending a huge amount of time cleaning the comments and recomputing their derivatives, such as negative/positive comment counts. Is this a useful step to take?

>> Resolved: Don’t use comments.

2. This dataset contains quite a lot of irrelevant videos (see the distribution of relevantTopicIds and weird discovery). My proposal is to remove a lot of videos based on Youtube's topicId labels. ARound 900 videos will be removed. What do you think of this approach?

>> TODO: Think of how to use later.

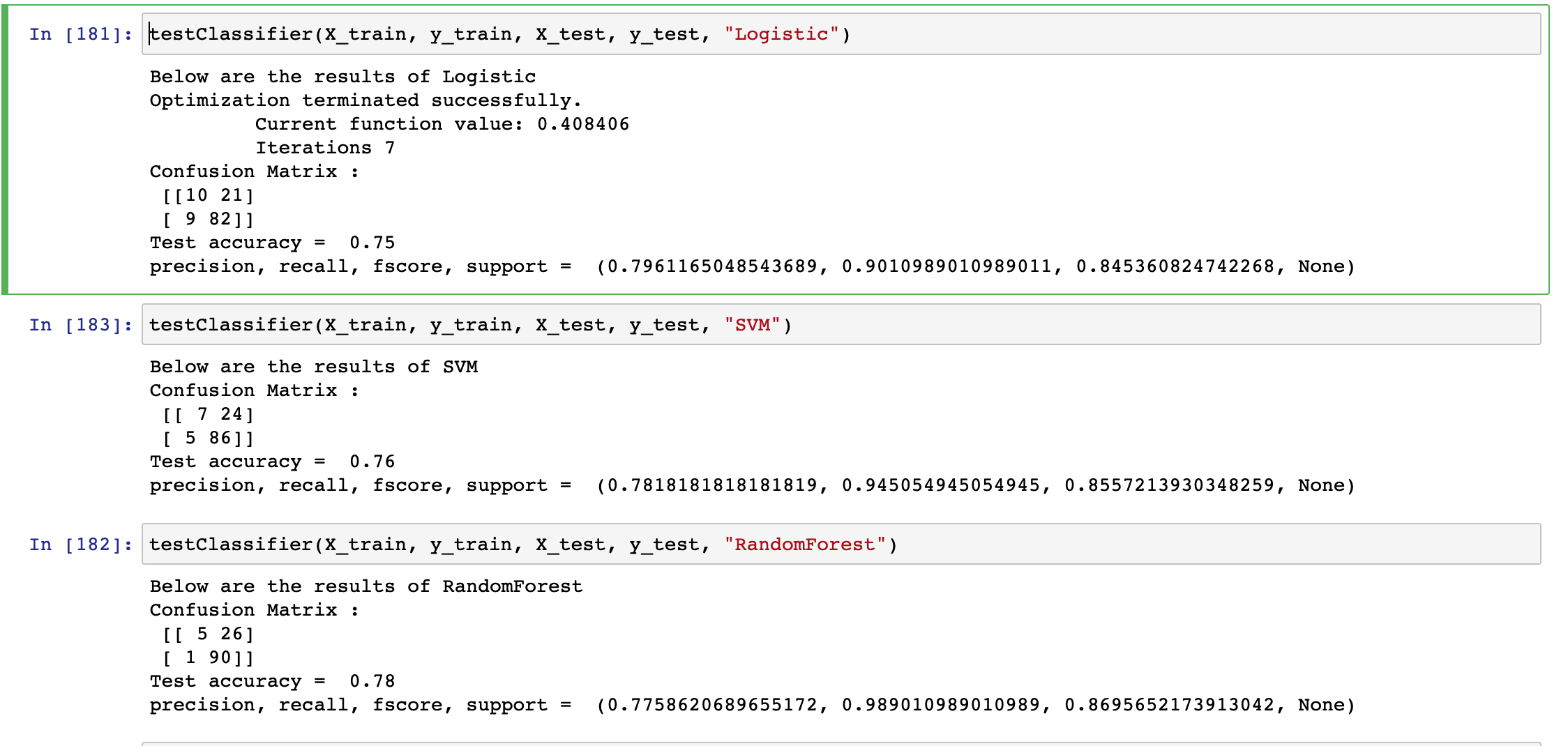
3. Many diabetes videos appears several times under different tags. These tags are definitely useful. However, my concern is that the co-training would label same video with different labels, especially on videos it doesn't predict with high confidence. How much will this affect co-training performance, if any? Any preventive measures necessary?

>> TODO: Later

4. I have compared three models: logistic regression, SVM, randomForest (see below). Overall, there’s negligible difference in accuracy where logistic regression performs slightly worse than SVM, randomForest, so it’s very unclear which model to choose based on these metrics.

What’s more concerning is that I could not replicate the results on Page 23 of Rema’s paper: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3711751>

My model includes fewer variables (i.e. no transcription confidence, no transcript medical term, no Boolean on tags/titles/description), so my model’s performance should be worse than what appears on the paper. The opposite happens.



I have set the seed and used approximately the same numbers in the evaluation set. Therefore, the difference in performance comes from something else (what?).

5. Co-training relies on the assumption that two sets of features are conditionally independent but can be used to predict equally well. There are a few videos where the description is exactly the transcript (from EDA), which definitely violates the assumption. Some transcripts contain absurd values (e.g., 19454) that shouldn’t be able to predict any labels. Your thoughts on how we might extend this?

6. Does co-training work with duplicates??

**## Extra stuff**

Distr

**## Major updates**

* 1. Xiao has cleaned the dataset.