Project Report for ENGG2112

Triggers and Symptoms of Chronic Autoimmune diseases

Mohsin Siddiqui, 520262652, Software Engineering
Bridgette Shore, 520510564, Chemical Engineering
Jamie Denovan, 510415103, Software Engineering

Faculty of Engineering

November 4, 2023

Executive Summary

Overall our goal is to create a more efficient way for patients to record, observe and tackle environmental triggers in order to avoid flares of chronic illnesses. The results from the model would be examined for dependencies and patterns, and thus allow better decision to be made using the saved data and trends. The pre-processing of the data was tough as it was a large, messy and disordered data file, which led to us limiting the scope to only the most occurring condition in the data-set. Our classifier models gave an unfortunately poor degree of accuracy, yet we were able to provide some insight upon the relative importance of general sets of factors upon fibromyalgia severity. The model has a high potential for improvement.

Contents

1 Project Overview

With recent data on global health, Machine Learning can be used to help patients of chronic autoimmune diseases and ‘invisible diseases’ and keep track of triggers in their environment that flare their symptoms, and if successfully applied, Machine Learning can have a significant impact on the health care system and thus society. In the USA, healthcare waste amounts to about 30 percent, of the 18 percent allocated to healthcare in the GDP. If even a fraction of these resources were used in different areas of the healthcare industry or other industries, it would transform the standard of living for millions of individuals.

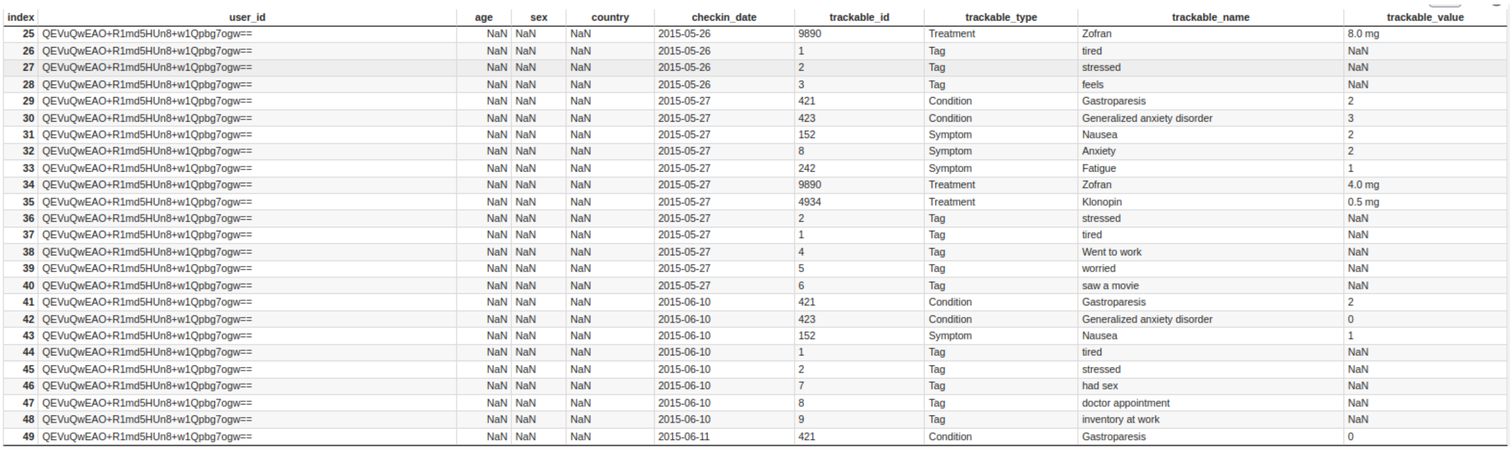
The goal for this experiment include, using the model to predict triggering factors when a certain condition is present in a user and or group of users. Thereby reducing dependency on physical tests and in-person physician appointments to record these flares. Making it more affordable for a significant group of people to look after their chronic disease symptoms without having to pay expensive fees at health practices. This reduces the risk of patients, especially the elderly and individuals from low-income groups of self-treating in isolation without any presence of medical supervision, and overall helps to improve the access to health education. This project collects data from patients into an app that then is able to track the severity of their symptoms and the potential triggers from the environment around them. Machine learning algorithms have the potential to track patients’ daily symptoms and assess external environmental factors and their potential correlation. This will assist in developing a personalised record of the individuals’ symptoms based on their experiences with trigger factors, and will potentially reduce medical costs, hospital resources, and the physicians time. This will provide doctors with another source of information to aid in treatment and relieving stress on the public health care system as resources can be redirected. If successfully implemented in a real-world scenario, this would help doctors save lives by allocating resources to patients who might need those more.

To make this project feasible, a range of data sets were researched including ones from Canvas, and concluded that our project would use data uploaded by Flaredown on Kaggle, named ’Chronic illness: symptoms, treatments and triggers’. The data is publicly accessible and is organised into 9 categories, ’User ID’, ’Age’, ’Sex’, ’Country’, ’Check in date’, ’Trackable ID’, ’Trackable type’, ’Trackable name’, ’Trackable value’. This data was originally used to design an app that would track the users symptoms, their current treatments and the environmental impacts. Fibromyalgia is a chronic disease with musculoskeletal pain, fatigue, sleep, memory and mood issues. Through the groups data analysis, Fibromyalgia was found to be the most prominent disease within the data set.

2 Methodology

2.1 Data Pre-Processing

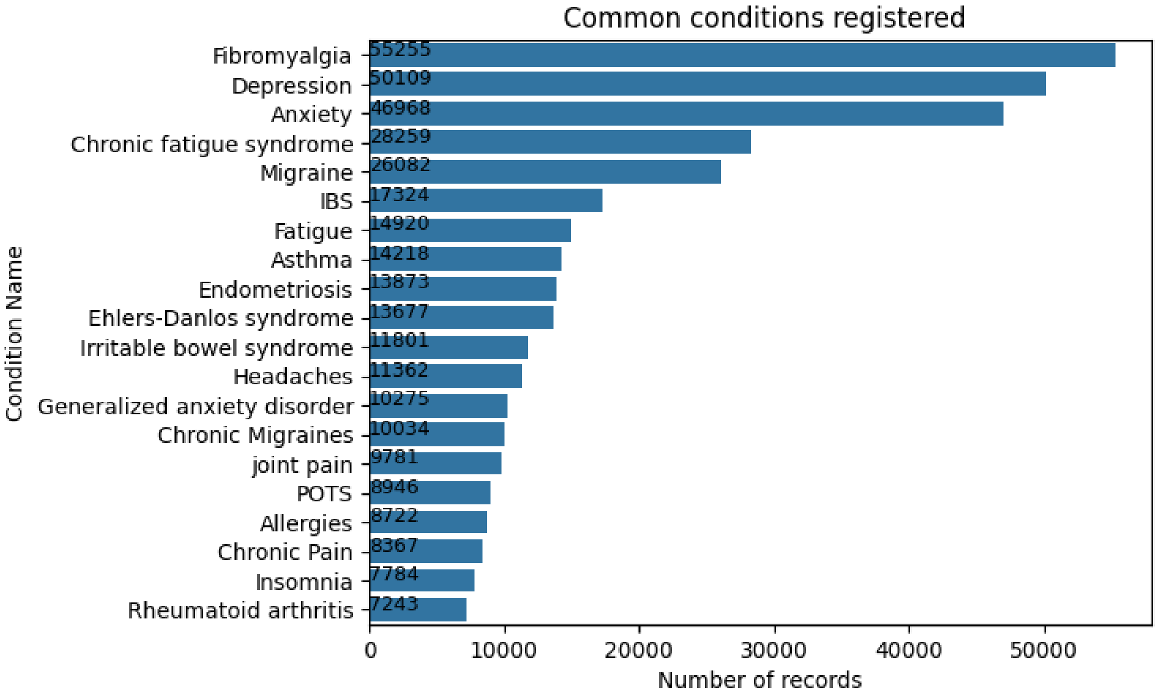
The data required significant pre-processing in order to reach a state suitable for analysis. An example excerpt of the raw data (of which there are 7976223 rows) is provided below.



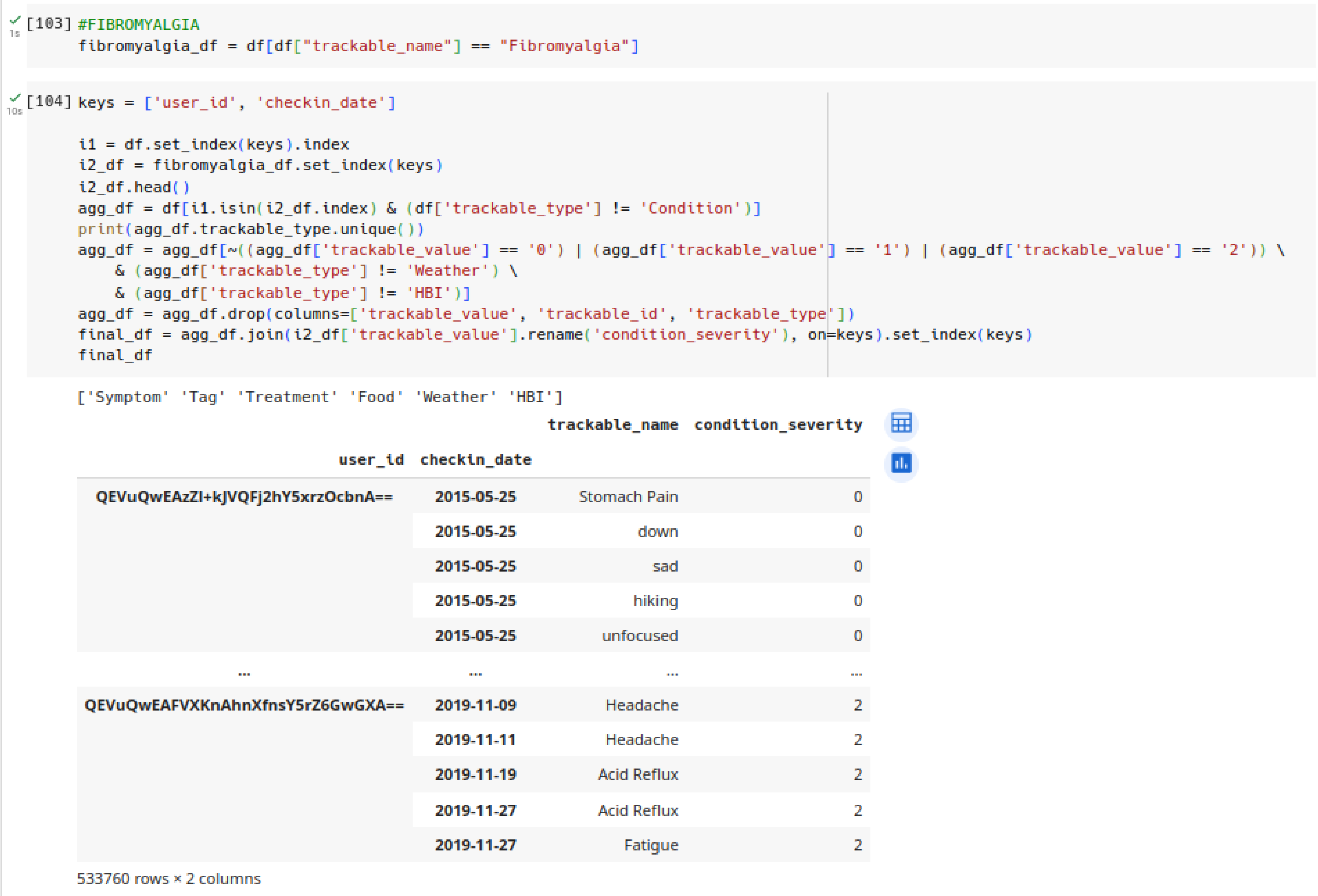
index	user_id	age	sex	country	checkin_date	trackable_id	trackable_type	trackable_name	trackable_value
25	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-26	9890	Treatment	Zofran	8.0 mg
26	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-26	1	Tag	tired	NaN
27	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-26	2	Tag	stressed	NaN
28	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-26	3	Tag	feels	NaN
29	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	421	Condition	Gastroparesis	2
30	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	423	Condition	Generalized anxiety disorder	3
31	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	152	Symptom	Nausea	2
32	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	8	Symptom	Anxiety	2
33	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	242	Symptom	Fatigue	1
34	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	9890	Treatment	Zofran	4.0 mg
35	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	4934	Treatment	Klonopin	0.5 mg
36	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	2	Tag	stressed	NaN
37	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	1	Tag	tired	NaN
38	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	4	Tag	Went to work	NaN
39	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	5	Tag	worried	NaN
40	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-05-27	6	Tag	saw a movie	NaN
41	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-10	421	Condition	Gastroparesis	2
42	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-10	423	Condition	Generalized anxiety disorder	0
43	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-10	152	Symptom	Nausea	1
44	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-10	1	Tag	tired	NaN
45	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-10	2	Tag	stressed	NaN
46	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-10	7	Tag	had sex	NaN
47	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-10	8	Tag	doctor appointment	NaN
48	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-10	9	Tag	inventory at work	NaN
49	QEVuQwEAO+R1md5HUn8+w1Qpbg7ogw==	NaN	NaN	NaN	2015-06-11	421	Condition	Gastroparesis	0

Figure 1: An excerpt of the raw data from the Flaredown CSV.

As our aim is to predict severity of an individual condition (using the provided categories of severity, a 0-5 scale) based upon a user's additional noted factors on a given day, we limited the scope of our analysis to fibromyalgia (the most commonly noted condition).



Since a predictive model requires a one-to-one mapping of input features and classification output, we performed split-filter-join operations upon our data frame to produce a mapping between miscellaneous trackables to Fibromyalgia severity for each given combination of (`user_id, checkin_date`).



```
[103] #FIBROMYALGIA
      fibromyalgia_df = df[df["trackable_name"] == "Fibromyalgia"]
```

```
[104] keys = ['user_id', 'checkin_date']

      i1 = df.set_index(keys).index
      i2_df = fibromyalgia_df.set_index(keys)
      i2_df.head()
      agg_df = df[i1.isin(i2_df.index) & (df['trackable_type'] != 'Condition')]
      print(agg_df.trackable_type.unique())
      agg_df = agg_df[~((agg_df['trackable_value'] == '0') | (agg_df['trackable_value'] == '1') | (agg_df['trackable_value'] == '2')) \
          & (agg_df['trackable_type'] != 'Weather') \
          & (agg_df['trackable_type'] != 'HBI')]
      agg_df = agg_df.drop(columns=['trackable_value', 'trackable_id', 'trackable_type'])
      final_df = agg_df.join(i2_df['trackable_value'].rename('condition_severity'), on=keys).set_index(keys)
      final_df
```

```
['Symptom' 'Tag' 'Treatment' 'Food' 'Weather' 'HBI']
```

		trackable_name	condition_severity
user_id	checkin_date		
QEVuQwEAzZI+kJVQFj2hY5xrzOcbnA==	2015-05-25	Stomach Pain	0
	2015-05-25	down	0
	2015-05-25	sad	0
	2015-05-25	hiking	0
	2015-05-25	unfocused	0
...	...	...	...
QEVuQwEAFVXKnAhnXfnsY5rZ6GwGXA==	2019-11-09	Headache	2
	2019-11-11	Headache	2
	2019-11-19	Acid Reflux	2
	2019-11-27	Acid Reflux	2
	2019-11-27	Fatigue	2

533760 rows × 2 columns

The fact that data such as symptoms, conditions, treatments and tags were input by users themselves necessitates further processing to standardize categorical values, as they created an opportunity for typos, a variety of abbreviations and different character choices, for example capital letters. This standardization was performed during the feature extraction stage.

2.2 Feature Extraction

Due to the extremely high cardinality of our noisy categorical data, it was immediately evident that one-hot encoding would be unsuitable for our purposes. Our research led us to the `skrub` library [1], which provides drop-in replacements for one-hot encoding on noisy, high-cardinality data which remove the need for manual feature engineering and data cleaning [2].

In particular, we used the `GapEncoder()` (Gamme-Poisson) transformer, which standardizes noisy categorical variables by instead encoding them as vector across a set of latent categories. These categories themselves are built by modelling a bag-of-n-grams representation of the input data **V** as a linear combination of topics **V** = **HW**, with **W** `(n_topics, vocab_size)` the topics (initialized via KMeans clustering) and **H** `(n_samples, n_topics)` the associated activations. [3]

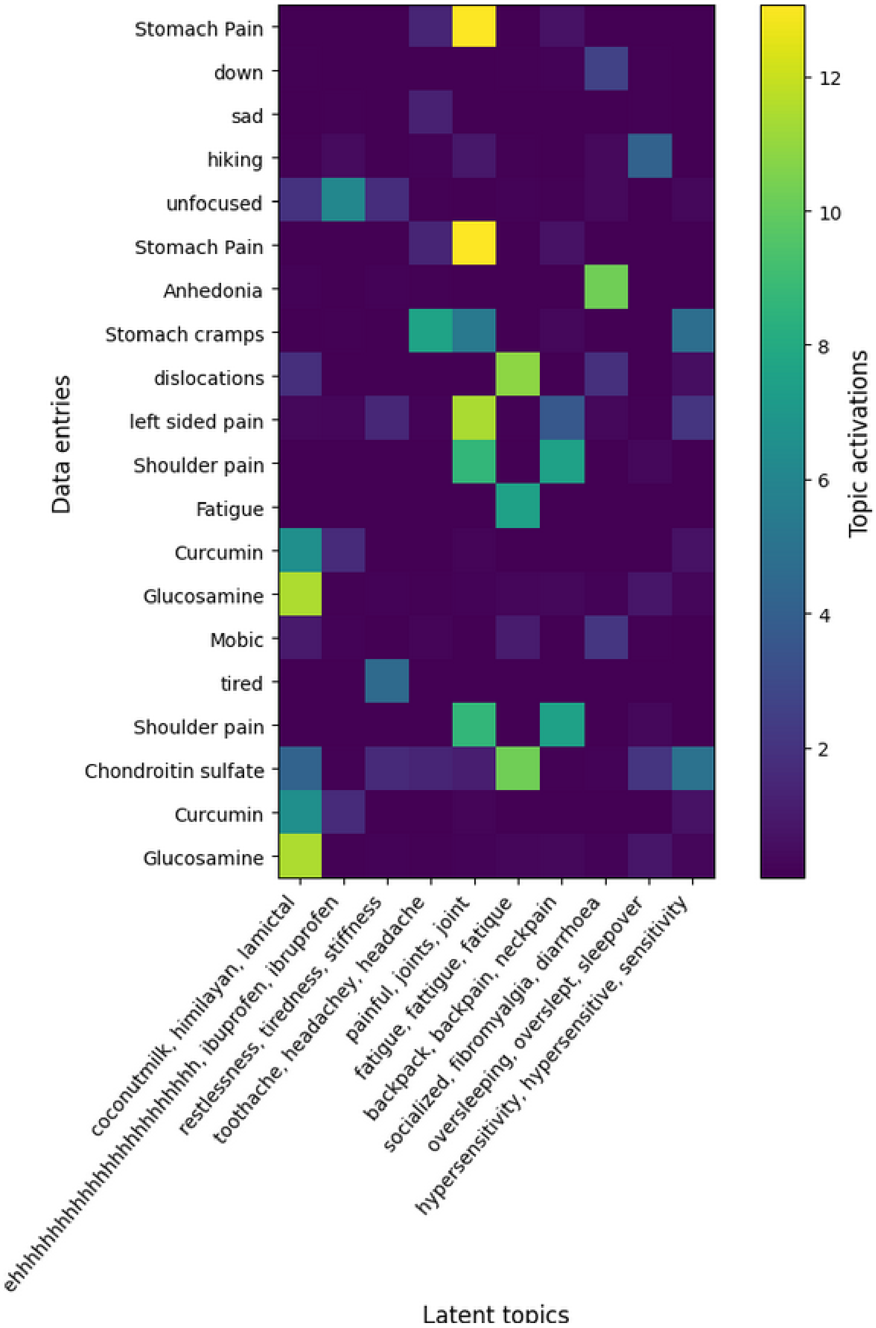


Figure 2: Representation of GapEncoder output on `n_components = 10`. Learnt topics are listed on the X-axis, with an example set of raw variables on the Y-axis.

2.3 Classification

As recommended in the documentation of skrub, we applied both the `RandomForestClassifier()` and `HistGradientBoostingClassifier()` models to the resultant output of the GapEncoding transformer. We systemically iterated through values of salient parameters for each model (`n_estimators` and `max_iter`, respectively) and calculated the cross validation score individually.

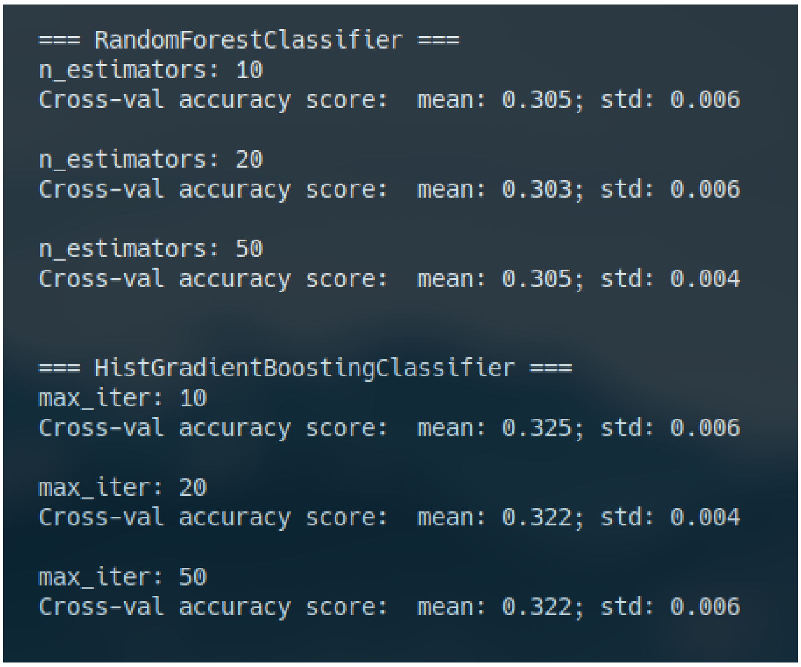
2.4 Simulation Environment

The simulations were run on Google Colab through Python notebooks. Cross-validation of 6 instances was performed, with the final model chosen through the highest-performance model. The metric used for evaluating this performance was mean accuracy, as provided by the classifier's `score()` methods.

3 Findings

3.1 Key Findings and Significance

Unfortunately, the output of our cross-validation gave universally poor performance with a mean accuracy of 0.3 across all models.



```
=== RandomForestClassifier ===
n_estimators: 10
Cross-val accuracy score:  mean: 0.305; std: 0.006

n_estimators: 20
Cross-val accuracy score:  mean: 0.303; std: 0.006

n_estimators: 50
Cross-val accuracy score:  mean: 0.305; std: 0.004


=== HistGradientBoostingClassifier ===
max_iter: 10
Cross-val accuracy score:  mean: 0.325; std: 0.006

max_iter: 20
Cross-val accuracy score:  mean: 0.322; std: 0.004

max_iter: 50
Cross-val accuracy score:  mean: 0.322; std: 0.006
```

Figure 3: Cross-Validation Score results across classifier model and parameter choice.

Nevertheless, we produced quantitative results by visualizing the 'feature importance' attribute of our `RandomForestClassifier()`, which produces a normalized estimate of the predictive power of a given feature [4], which in our case refers to the latent categories uncovered by the `GapEncoder`.

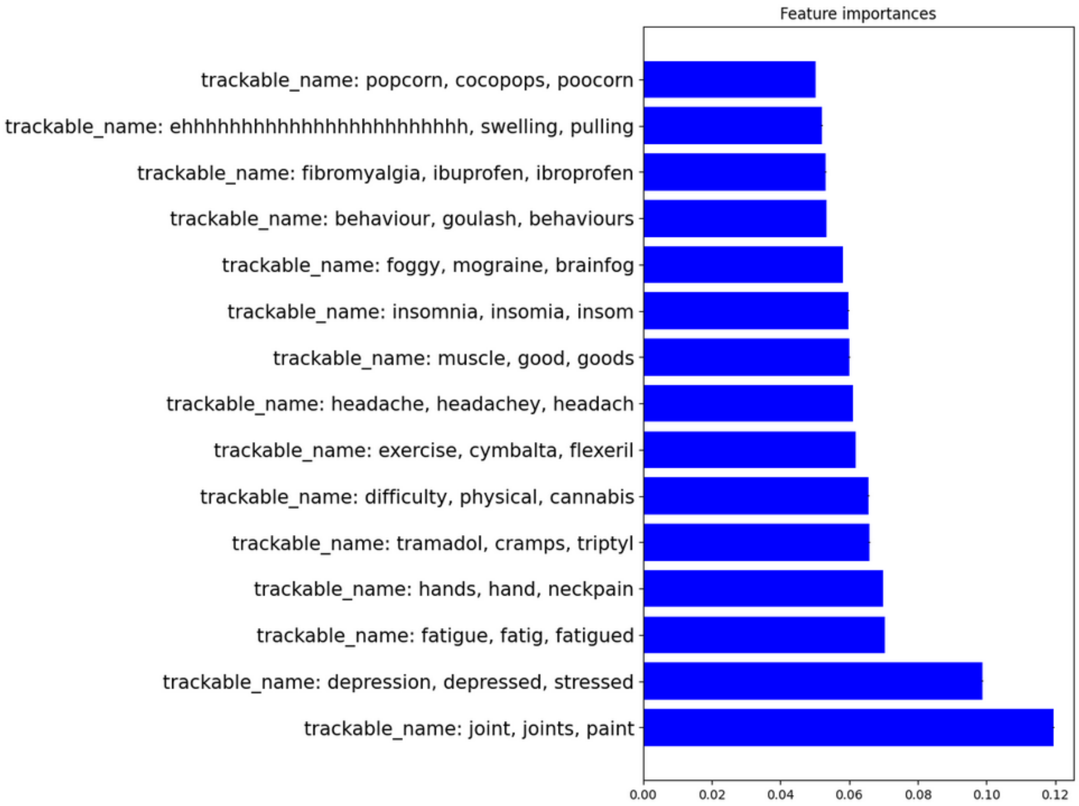


Figure 4: Representation of the predictive importance of individual features.

As could be expected, joint pain had the highest feature importance. But the second highest feature importance was of **depression/stress**, which is a much less recognized symptom of Fibromyalgia (ref). For a lot of time, the general view of Fibromyalgia has been that it wasn't a *real* medical condition. Now our study has found depression and stress to be the second leading symptom that increases severity of Fibromyalgia. This shows the model successfully found a symptom that was otherwise overlooked for the condition, and if this led to doctors focusing more on the mental health aspect of fibromyalgia, people may require less hospital visits for treatment, which would reduce the strain on health facilities.

3.2 Issues Faced

The coding of the simulation was not without a couple of hurdles. We faced the problems that follow in order of listing, and were dealt with as written.

- **Problem One**: The format of the source CSV was unsuitable for analysis - input and target variables were placed under the same columns, in separate rows.
- **Solution**: A separate dataframe was created, filtered to the rows containing target scores. These rows were then joined against the set of rows containing input data along the multi-index keys of user id and date.

- **Problem Two**: User-input meant typographical errors, morphological representations, etc in the data, which meant it was hard to classify the categorical variables. **Solution**: GapEncoder function in the Skrub package created encodings of vectors across a set of latent categories, evaluated by fitting correlations between substring tri-grams.

4 Potential for Wider Adoption

There isn't a lot of software currently that would help solve the issues our project addresses, and the ones that do exist, usually have a low accuracy rate between 19-38 percentage (ref). A accuracy that low wouldn't be acceptable in a medical practice. Our model wasn't perfectly accurate, but does have a lot of room for improvement with the suggestions given below. Our project would reduce the cost of existing systems and practices in the medical industry which would mean more funds allocated towards areas our project wouldn't cover as a generally accepted tool in practice. For commercialization: designed models that would be marketed to health practices to be utilized by their practitioners in their work, not dissimilar to pharmaceutical companies. *There is also a scope for software related to tracking itself, since the pre-existing ones do not quite deliver the data collection experience that would be paramount to make this project a payoff commercially.*

We envisage that future work might include:

- Improvement One: Improving the accuracy of the model to a minimum of better than half by testing with more 'clean' data and more number of conditions taken into account. This would mean it would be more trusted to be utilized by health staff/practitioners.
- Improvement Two: Our findings currently list the most prevalent symptoms and hence the next step, would be for the model to be able to discern on its own, how the symptoms correlate to the severity of the conditions, how already relevant they are in discussions and treatment involving the condition to a higher degree of accuracy.
- Improvement Three: The depths of insights available (and thus, potentially the accuracy of our model) could be improved by first aggregating the GapEncoded features along user & date in a manner similar to using a `MultiLabelBinarizer`. However, combining the multidimensional output of the `GapEncoder` with a similar processor was beyond the scope of our abilities.
- Adjustment one: The data collection should be better at cleaning the data by perhaps setting the data entries for symptoms to pre-set list of words.

5 Conclusions

The project was completed with a limited degree of success. We managed to transform the raw data into a format fairly suitable for analysis, although we encountered significant difficulty in attaining high accuracy with our classifier models. The main findings were as follows:

1. Has potential to identify and highlight symptoms not closely followed and monitored in treatment of a condition.
2. Joint pain and depression were found to have relatively high and almost equal importance on fibromyalgia severity.
3. Further refinement needed to improve predictive accuracy.

Our project's impact could be improved by sourcing more clean and efficient data. With a low accuracy, our project would need a more reliable model before being implemented on a wide scale in the health industry, but could still be trialed in smaller environments under observation.

References

[1] "skrub." `https://github.com/skrub-data/skrub`, 2023.

[2] P. Cerda and G. Varoquaux, "Encoding high-cardinality string categorical variables." working paper or preprint, July 2019.

[3] "skrub api reference." `https://skrub-data.org/stable/generated/skrub.GapEncoder.html`, 2023.

[4] "Scikit-learn documentation."