

Model Card for README: Not an experience

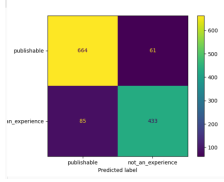
Data Science

Exported on 04/30/2024

Table of Contents

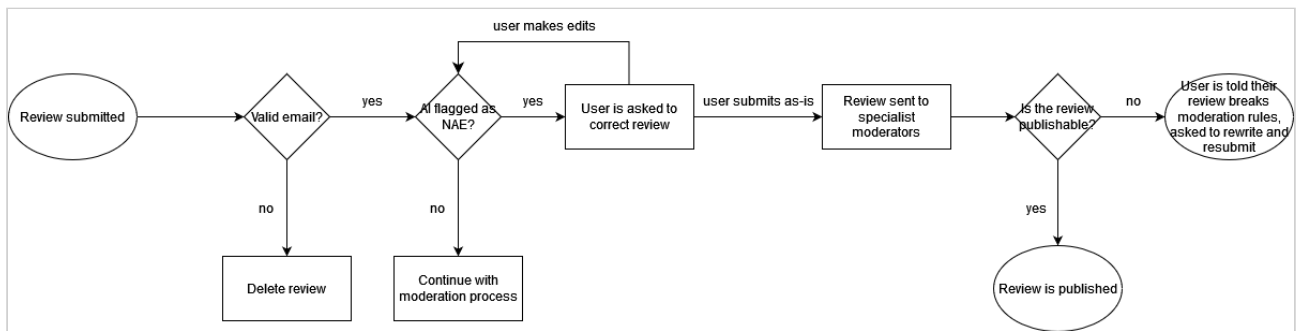
1	Model At A Glance:	3
1.1	Intended Use	4
1.2	Inputs and Outputs	4
2	Data	5
2.1	Training Data	5
2.2	Evaluation Data	5
3	Methodology	7
3.1	Training methods	7
3.2	Model evaluation	8
3.3	Quantitative Analyses	8
3.4	Ethical Considerations	9
3.5	Caveats and Recommendations	9
4	Model Deployment	11

1 Model At A Glance:

Description :	The model detailed in this model card was designed to moderate reviews submitted to the nhs.uk ratings and reviews (RAR) service. It is a classifier model that analyses the review text, and it distinguishes detailed personal experiences with NHS providers from generic comments that do not describe a specific experience ("not an experience" - NAE).				
Model Type:	Natural language processing (NLP), using supervised logistic regression classification				
Developed By:	NHS England's Data Science Skilled Team (@Liliana Valles Carrera and @Alice Tapper)	Confusion Matrix:		F1*:	0.856
Launch Date:	24 Jan 2024 tbc			False positive (FP) rate:	8.4%
Version:	1.0			False negative (FN) rate:	16.4%

- Development background:** This model is one of a collection of models that have been designed by NHS England's Data Science Skilled Team (DSST) for the nhs.uk ratings and reviews (RAR) service, to replace human moderation of incoming reviews. The models have been designed to implement [RAR policy](https://www.nhs.uk/our-policies/comments-policy/)¹, which defines a list of rules which reviews must pass in order to be published on the website. This model relates to the frequent references to the user's **experience**, i.e. the review should relate to a single experience with the NHS provider, and not a more general review of the service. If the model flags the review as NAE, the user will be asked to either make appropriate edits to the review and resubmit, or submit the review as-is to specialist moderators for further review. If the model does not flag the review as NAE, the review will continue through the moderation process, and potentially be autopublished to the website if none of the other models flag issues.

¹ <https://www.nhs.uk/our-policies/comments-policy/>



1.1 Intended Use

- **Scope:** Distinguishing detailed personal experiences with NHS providers from generic comments that do not describe an experience.
- **Intended Users:** NHS patients who wish to leave a review on nhs.uk RAR service about their experience with NHS providers; NHS.UK RAR Team who will use the result from the model to make certain autopublishing and moderation decisions.
- **Use Cases Out of Scope:** Classifying reviews or comments in any other context

1.2 Inputs and Outputs

- **Input:** A review submitted to the nhs.uk RAR service. The current endpoint where the model is deployed takes the input format as a json encoded dictionary:

Input Data

```
{"data": [review text]}
```

- **Output:** A 0/1 classification indicating whether the review passes/breaks the NAE rule, alongside the classification probability. The current endpoint outputs these results also as a json encoded dictionary:

Output Data

```
{"0": classification label, "1": classification probability}
```

This classification will feed into any guidance that the user is shown, if it is necessary for them to revise their review.

2 Data

- **Data overview:** The model uses text data primarily sourced from user reviews about healthcare providers submitted on the [NHS.UK website](https://www.nhs.uk/contact-us/find-out-how-leave-review-of-nhs-service/)². These reviews, stored in Dynamics CRM for up to two years, were subject to a rigorous two-level moderation process involving both external and internal teams. Due to historical inconsistencies in moderation, the internal team had to analyse this stored data and curate it to ensure accuracy and consistency. Several of these curated datasets were provided, containing a mix of reviews: those that break the NAE rule (2083), and those that were published without issues (2886). This dataset collection was combined with an additional training-only dataset of published reviews (3246, sourced from the RAR service but not subject to checks and curation, see **training data** below) and provides a comprehensive range of user feedback, crucial for training the model and evaluating its performance.
- **Data pre-processing and cleaning:** As previously stated, the internal moderation team checked the datasets to guarantee the examples of NAE reviews and publishable reviews were accurate. There was one exception, mentioned in **training data** below. The datasets were also cross-checked for duplicates, and any that were found were removed.
 - **Encoding model:** To encode the text data for classification, a bag-of-words representation was used, where the 10,000 most common words (including stopwords) in the training data were included.
- **Split:** The provided data was split 75/25 into training/validation data. Some additional training-only published data was selected separately, details in **training data** below.

2.1 Training Data

- **Data Description:** The provided datasets of NAE reviews and published reviews were each split 75/25 into training and validation datasets. See above for information about data sourcing, re-processing, cleaning and encoding. The only exception to this standard data curation and 72/25 split was an extra dataset of published reviews. These reviews were selected by running a large dataset of unchecked, published data (~80,000) records through one of the first iterations of the NAE model. Only the reviews that were classified as publishable with a significantly high probability were selected. Due to the lack of checks by the internal moderation team, these reviews were used as training data only.

2.2 Evaluation Data

- **Datasets used:** The provided datasets of NAE reviews and published reviews were each split 75/25 into training and validation datasets. See above for information about data sourcing, pre-processing, cleaning and encoding.

² <https://www.nhs.uk/contact-us/find-out-how-leave-review-of-nhs-service/>

- **Why these datasets:** The validation data are all examples of real NAE/published reviews, so will give us an accurate representation of how the model will perform in live deployment
- **Representativeness:** The data has been checked thoroughly by the internal moderation team to make sure that they are all accurate, representative examples of NAE reviews. The dataset was judged large enough and varied enough to state the model performance with a reasonable degree of certainty.

3 Methodology

- **Model type:** Natural language processing (NLP), using supervised logistic regression classification
 - **Models used:** A bag-of-words representation is used to vectorize the review text, and a logistic regression model is used for classification
 - **Algorithm details:** First the review text is encoded using a bag-of-words representation. The result is a vector of length 10,000, listing the word counts in the review for the 10,000 most common words in the training data. This encoded text is then the input to the logistic regression classifier model, which outputs the final classification and corresponding probability. Post-training, the final logistic regression model parameters are: {C: 0.01, penalty: l2, solver: liblinear}. The classification probability threshold was 0.5.
- **Feature engineering:** For NAE rule, the encoded review text was used as the only feature. There was an attempt to break this text down into simpler features (see **alternative models explored** below) but this method did not outperform using the complete embedded review text.
- **Training process:** The logistic regression model was trained by fitting the model to the labelled training dataset
- **Hyperparameter tuning:** Cross-validation on the training data was used to select the best parameters for the logistic regression model, optimising the f1 score. 5 folds were used.
- **Alternative models explored:** Combinations of embedding models (BAAI/bge, intfloat/e5-large-v2, thenlper/gte) or other representations (bag-of-words, TFIDF, LDA) with classifiers (logistic regression, SVM, XGboost) were tested. Various methods of stacking these representations and classifiers were tested, as well as a voting model to combine the results of multiple models. The final model was selected because it outperformed these alternative combinations and methods. An investigation was also performed to explore how different, simpler features of the text (bigram frequency, tense, pronoun usage, sentiment and length of comments) varied between NAE reviews and publishable reviews. However, models based on these simpler features did not give significantly successful results.

3.1 Training methods

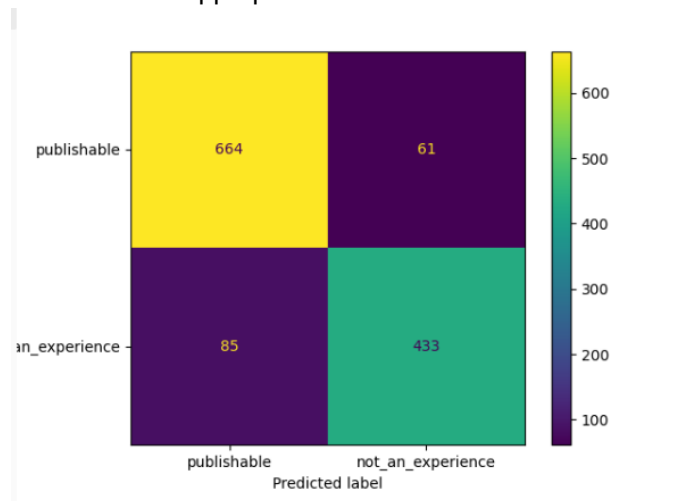
- **Literature review:** The definition of 'not an experience' used in this model is particularly bespoke, so a literature review was not deemed helpful/necessary.
- **Preferred approach:** Bag-of-words representation was included in the testing due to its simplicity and lightweight computation. Logistic regression was included in the testing because it is a standard, again fairly lightweight classification model. This combination was selected for the final model over alternative embeddings models/classifier combinations purely due to its superior performance.
- **Encoding and fine-tuning details:** The standard bag-of-words representation was used (with stopwords included), and a standard cross-validation on the training data using 5 folds as performed

3.2 Model evaluation

- **Model validation:** The model was evaluated by measuring its performance on the labelled validation data.
- **Evaluation focus:** To maximise overall performance of the model, the development and selection of a model that has a high f1 score was prioritised. Where there was a trade-off between the FP rate and FN rate, preference was shown towards a slightly lower FP rate, to try and avoid overburdening the internal RAR moderation team (as some of the reviews will still have to be internally moderated if the user contests the classification).

3.3 Quantitative Analyses

- **Performance breakdown:** Overall, the model is better at correctly classifying published reviews vs NAE reviews. This should avoid overburdening the internal moderators, however it does risk diluting the quality of the reviews published on the service. It should be noted here that the external moderators being replaced by the AI also make a not-insignificant number of mistakes classifying NAE reviews. Therefore the ability of the model to classify NAE reviews was deemed acceptable.
- **Model Validation:** The results of the model for the validation data are shown here using a confusion matrix and the appropriate metrics



Metrics:

- f1* score: 0.856
- FP rate: 8.4%
- FN rate: 16.4%

*N.B. The f1 score will vary based on the balance of the validation dataset. We chose to use all the validation data we had access to, in order to gain the most evidence about the model performance; as a result, the validation data here is both unbalanced and not representative of the true data split. In reality, there will be significantly more publishable reviews than NAE reviews. This means the f1 score

for 'real-world' performance will differ from the value shown here. This was considered appropriately when selecting the final model.

- **Bias and Fairness Analysis:** Bias analysis was performed for three of the models, including NAE. The results for the NAE model showed that the TP (true positive) reviews were significantly shorter than the FN reviews, and that the FP reviews were significantly shorter than the TN (true negative) reviews. This suggests that the model is more likely to classify a review as NAE if it is shorter. This could be due to various factors, such as the model's inability to maintain context over longer sequences. There is also a potential bias towards more accurately classifying not-an-experience reviews that have a positive sentiment compared to those with a negative sentiment. This could be due to sentiment biases in the training data.

3.4 Ethical Considerations

- **Sensitive data use:** All the data which users submit gets held in a Microsoft Dynamics 365 database. The reviews themselves can contain PID (e.g. people's names), and information about the reviewer (email, IP address) is also held as part of the record. [By submitting their review for publication, the user consents for their record to be used in analysis to improve the service.](#) These records are automatically deleted or redacted after a 2 year period. Note that in order to train our models, we have to export this data and import it into Azure Machine Learning Studio (AMLS). This does **not** have the same automatic deletion policy in place. In principle, it would be simple enough to create our own retention policy enforcer inside AMLS. However, this has not been done at time of writing, and indeed we have not thoroughly explored the extent to which the policy applies in the adapted context. Coming to a clear answer on this and implementing a solution is one of the outstanding tasks to accomplish. It is relevant to note here that unnecessary personal information which is not used for model training (such as user email, IP address) is not imported into or stored in AMLS.
- **Implications for human safety:** The 'not an experience' rule does not touch on any obviously sensitive topics. A user may become frustrated if they submit a review that is misclassified as 'not an experience'. However, this frustration is unlikely to cause them significant emotional distress or implicate their personal safety.
- **Potential harms:** Erroneously publishing 'not an experience' reviews might make the published reviews less helpful, but does not have any obvious ethical implications. As above, if a user is erroneously informed that their review is 'not an experience', this may be frustrating. To counter this for the initial model launch, if the user is told their review breaks the NAE rule and they disagree, they will still have the option to send the review for moderation by the internal moderation team, which should reduce their frustration.

3.5 Caveats and Recommendations

- **Caveats and Limitations:**
 - Bag-of-words can only represent word frequencies for words included in the training data. A large, representative training dataset was used to account for this, but it is still possible for new reviews to use words not included in the training data. To counter this, retraining is

advised if language changes over time or if new slang or terminology emerges, e.g. the word "covid" appearing during the COVID-19 pandemic.

- **Future directions:**

- Further investigation into the bias analysis to see if there are steps that can be taken to reduce model bias
- Possible further exploration of a voting model (combining the results of a non-semantic approach and a semantic, embeddings approach)
- Tuning the classification threshold to effectively bias the model towards false positives or false negatives
- Deployment of the model to a smaller compute, following analysis to see whether this would affect overall performance of the automoderation tool

- **Additional notes:**

4 Model Deployment

- **Versioning:** The version of this model is v1.0.
- **Links:**
 - Model training found at `driving_scripts > model_training > register_nae_model.py`
 - Model deployment script along with score script found under `driving_scripts > deploy_scripts > deploy_nae`
- **Inference strategy:** The inference process for the NAE model involves receiving and processing user reviews in real-time through a Flask app, which queries the Azure-hosted model endpoint. The model first translates the text to bag-of-words representation, and then performs the classification task. The classification is returned to the Flask app, alongside the corresponding probability. No further post-processing occurs on the Flask App.
- **Deployment strategy:** The model, developed on Azure Machine Learning, leverages Azure's managed services. This means the compute resources are managed by Azure, simplifying scaling and management processes. The models are deployed using the Standard_DS3_v2 SKU, a general-purpose endpoint type that provides a balance of CPU, memory, and I/O, suitable for various workloads. This SKU features 4 cores and 14 GiB of RAM. The deployment allows for both manual and automatic scaling, with autoscaling as a built-in feature. The current setup, with an instance count of 1, is optimized for the anticipated workload but can be adjusted based on demand. Monitoring performance and response times dashboards were set up to determine the need for scaling, either by increasing the number of instances or upgrading to a more powerful SKU. The endpoint is secured with key-based authentication to ensure only authorized access.
- **Model performance in deployment:** There are no results post-deployment at the time of release, since this model was not dark-launched before going live. Going forward, [ongoing monitoring and evaluation are in place to ensure the model's performance remains consistent and reliable, with periodic updates as needed.](#)
- **Timing analysis:** Some general timing analysis was performed to see how quickly the Flask app is able to query, receive and handle the results from each endpoint. This can be used as an indirect measure of the endpoint latency for practical purposes. The NAE model was the fastest of the various moderation endpoints, with the Flask app taking an average of 0.1s to send and receive the NAE classification per review. Since the various moderation endpoints are queried in parallel, this means that the performance of the NAE model is not a limiting factor to the speed of the automoderation processing. Further to this, user testing of the review submission experience was completed and found the speed of the automoderation processing to be acceptable.