Topic Modeling and Sentiment Analysis of User Reviews

Latent Dirichlet Allocation (LDA) Model

Project Members

Sajal Debnath
Krishanu Chattopadhyay

November 22, 2024

Project Description

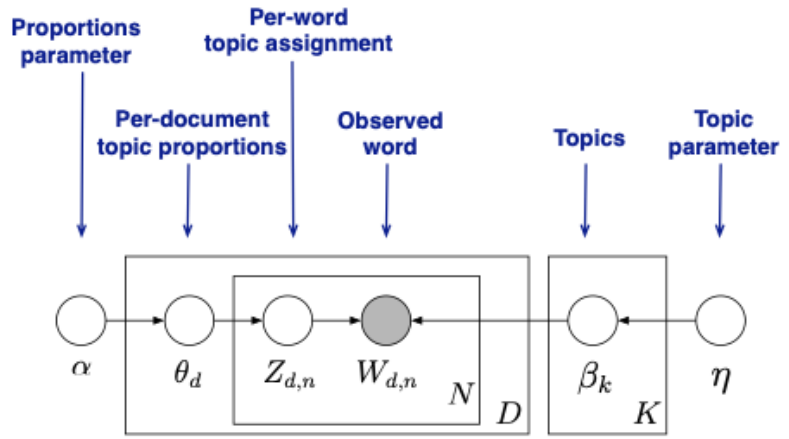
Topic Modeling and Sentiment Analysis of User Reviews

It is very important for any organization in today's world to know the feedback and customer sentiments about their business generally in the form of reviews. This is more true for service industries like hotels, restaurants to name a few. But with the numbers being in thousands and lakhs, it is humanely impossible to read all the feedbacks and understand/co-relate the sentiments.

We are trying to solve this problem through machine learning models. Where we want to create a model which not only can go through all the reviews but divide them into topics and then find out the general sentiments on them.

So, we are aiming to do topic modeling which is used to discover hidden themes and patterns within large collections of text data, allowing researchers and analysts to understand the underlying topics discussed in a corpus of documents without manually reading each one, essentially providing a way to organize and make sense of large volumes of unstructured information by identifying recurring themes and keywords across the data.

LDA as a graphical model



- Nodes are random variables; edges indicate dependence.
- Shaded nodes are observed; unshaded nodes are hidden.
- Plates indicate replicated variables.

Topic Modeling using LDA

- LDA performs text analysis with probability models.
- LDA discovers themes using posterior inference.
- LDA helps us discover hidden themes in documents.

How does LDA Work?

- LDA trades off 2 goals
  - In each document, allocate its words to few topics
  - In each topic, assign high probability to few terms.
- These 2 goals are at odds
- Trading off these goals finds groups of tightly co-occurring words.

Trip Advisor Dataset, Data cleanup and Data preprocessing methods

Dataset

Review data downloaded from kaggle

– Trip Advisor Hotel Reviews

Data cleanup

Used regular expression to remove

– Links, Mention symbol, Hashtag, Punctuations

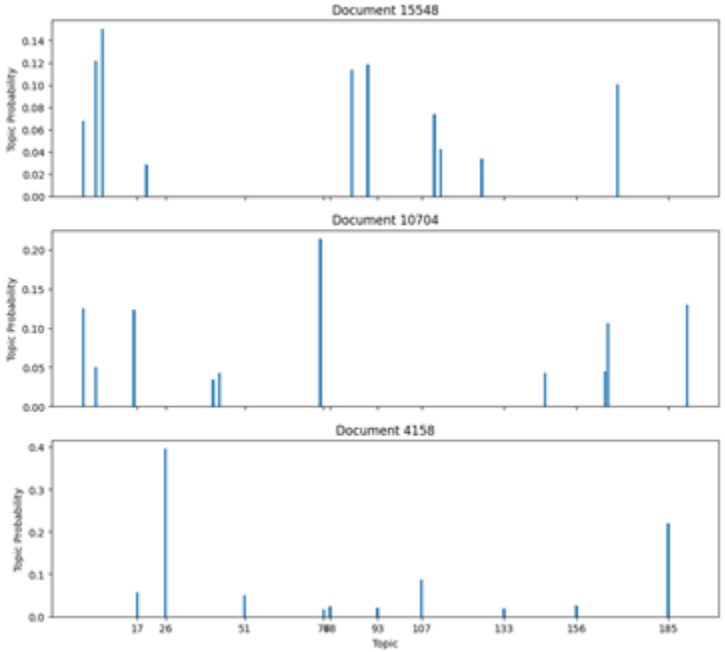
Data preprocessing

– Lower casing

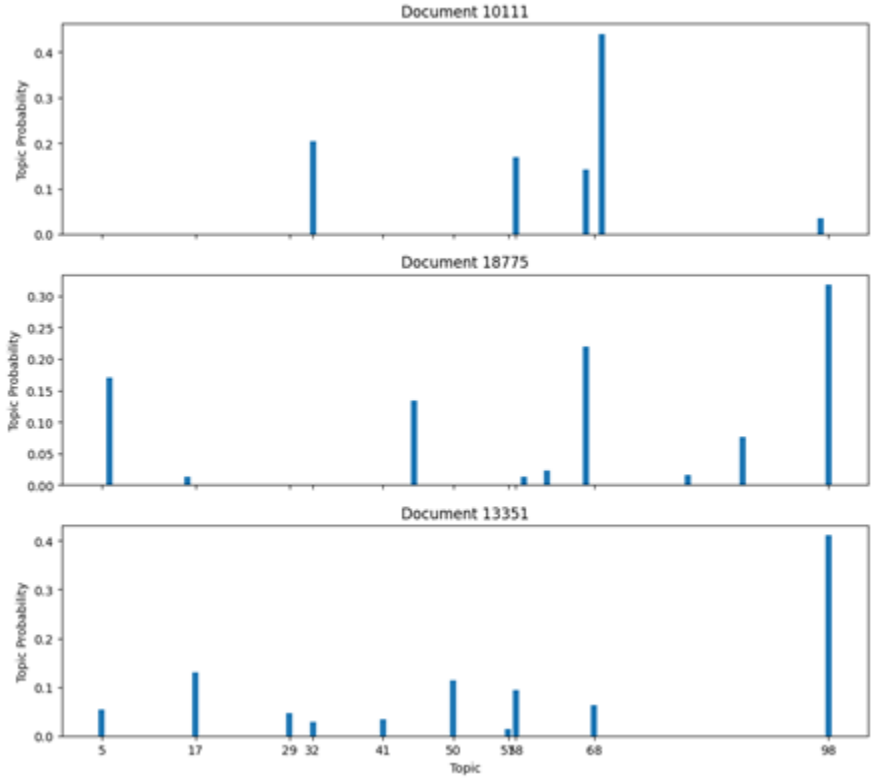
– Stop word filtering

– Lemmatization

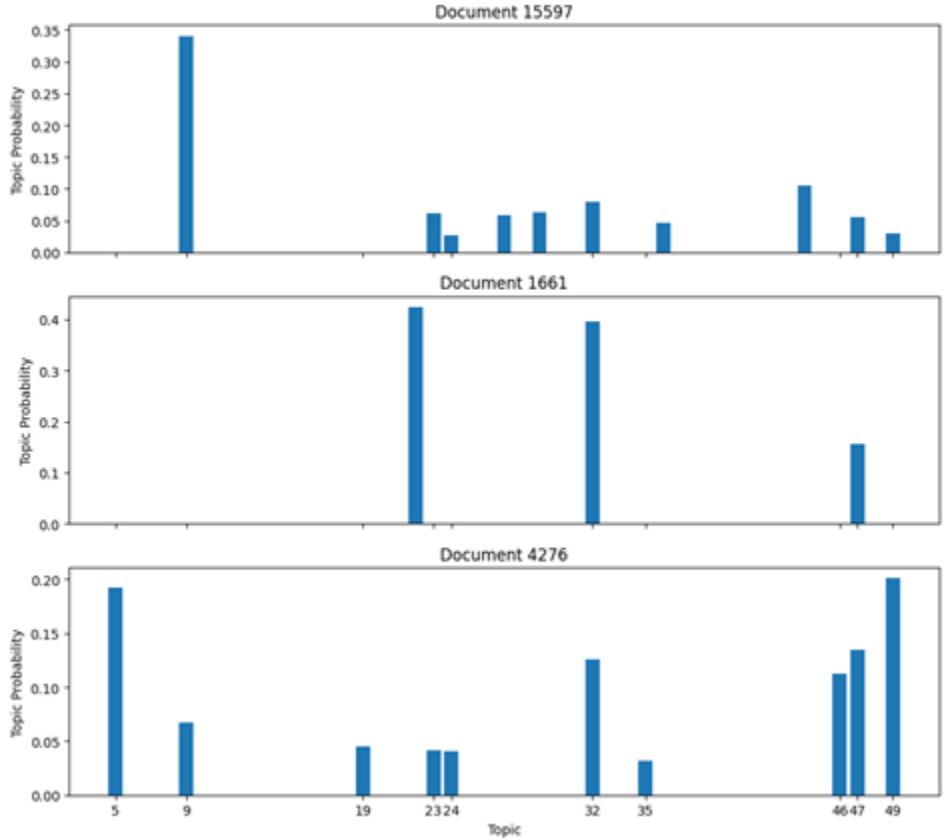
Per document topic proportions



Number of topics = 200



Number of topics = 100



Number of topics = 50

Per topic word distribution

Words from 20 sample topics

(**27, '0.022***"room" + 0.014*"beijing" + 0.014*"majestic" + 0.012*"great" + **0.008***"staff" + 0.008*"stayed" + 0.007*"view" + 0.007*"bed" + 0.007*"stay" + **0.007***"night"')

(**41, '0.017***"great" + 0.014*"location" + 0.012*"stay" + 0.010*"gallery" + **0.010***"breakfast" + 0.009*"walk" + 0.009*"academy" + 0.008*"staff" + 0.008*"nice" + **0.008***"room"')

(**37, '0.026***"room" + 0.011*"time" + 0.010*"good" + 0.008*"pool" + **0.008***"staff" + 0.007*"great" + 0.006*"day" + 0.006*"2" + 0.005*"restaurant" + **0.005***"stay"')

(**45, '0.137***"barcelona" + 0.097*"euro" + 0.026*"fab" + 0.011*"day" + **0.009***"city" + 0.009*"great" + 0.008*"night" + 0.008*"stay" + 0.007*"room" + **0.007***"la"')

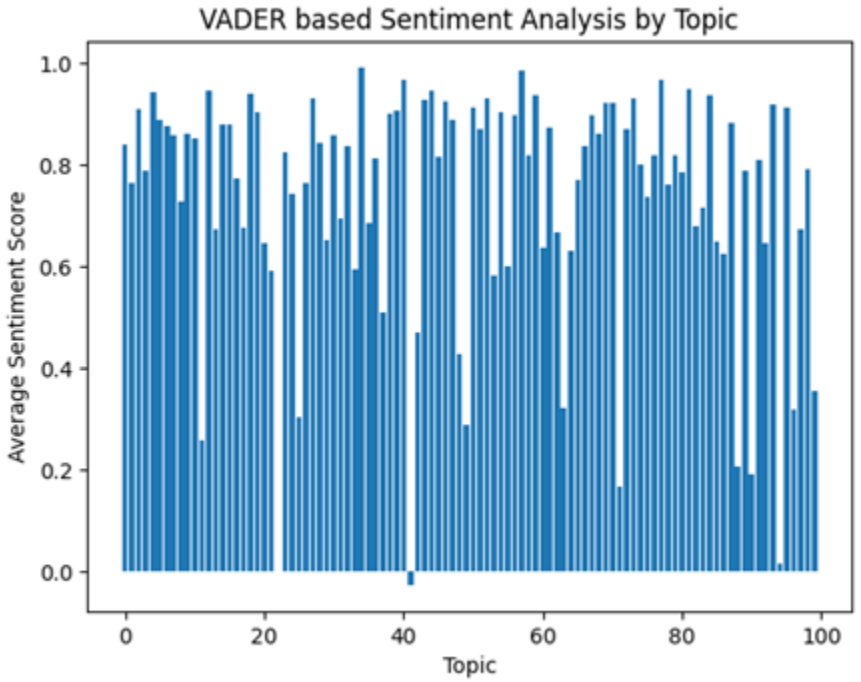
(**3, '0.036***"punta" + 0.032*"cana" + 0.021*"riu" + 0.021*"resort" + **0.018***"horse" + 0.013*"people" + 0.012*"food" + 0.011*"beach" + 0.009*"awsome" + **0.008***"6star"')

(**35, '0.015***"resort" + 0.014*"villa" + 0.011*"beach" + 0.010*"great" + **0.010***"time" + 0.010*"cana" + 0.010*"punta" + 0.010*"pool" + 0.009*"food" + **0.009***"room"')

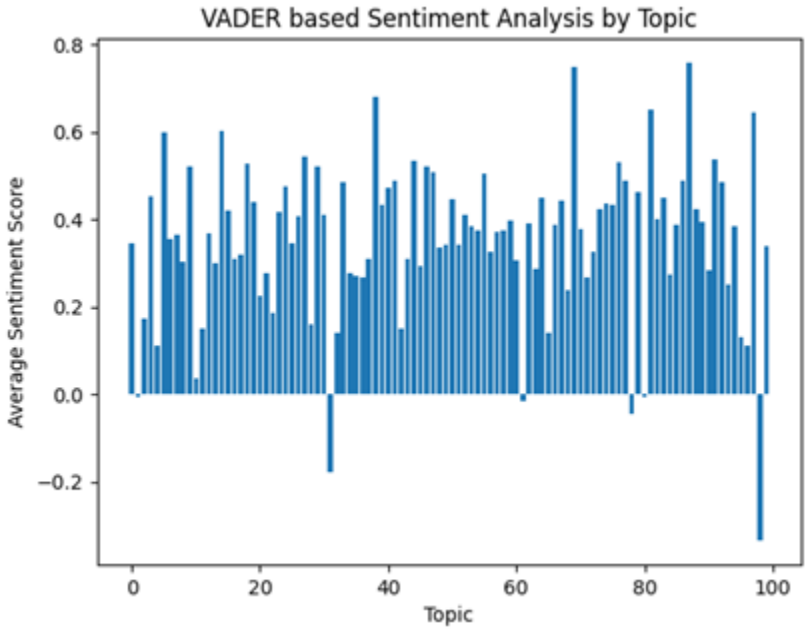
(**24, '0.039***"great" + 0.021*"location" + 0.018*"room" + 0.017*"stay" + **0.015***"good" + 0.014*"staff" + 0.013*"nice" + 0.012*"clean" + 0.010*"place" + **0.010***"walk"')

Sentiment Score (VADER method)

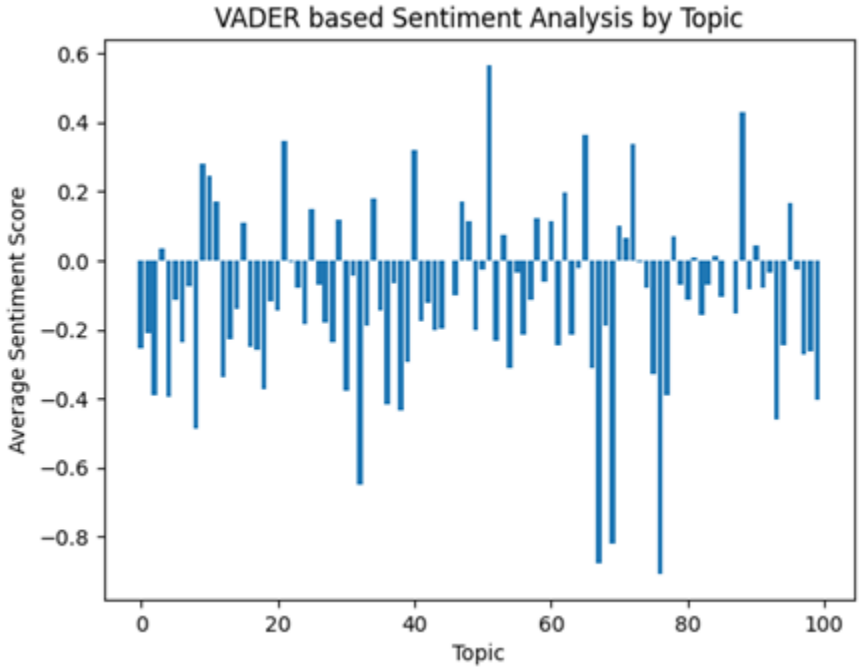
Lower ratings show negative sentiment



Ratings 1 to 5



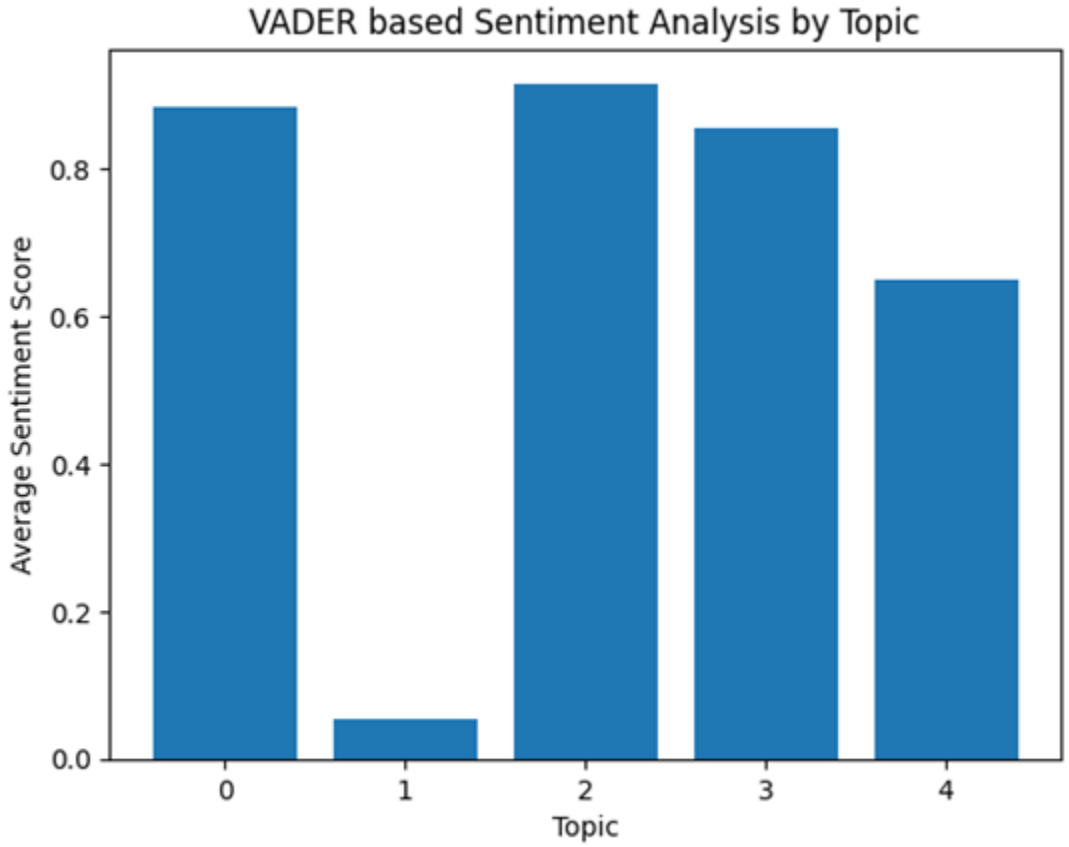
Ratings 1 to 3

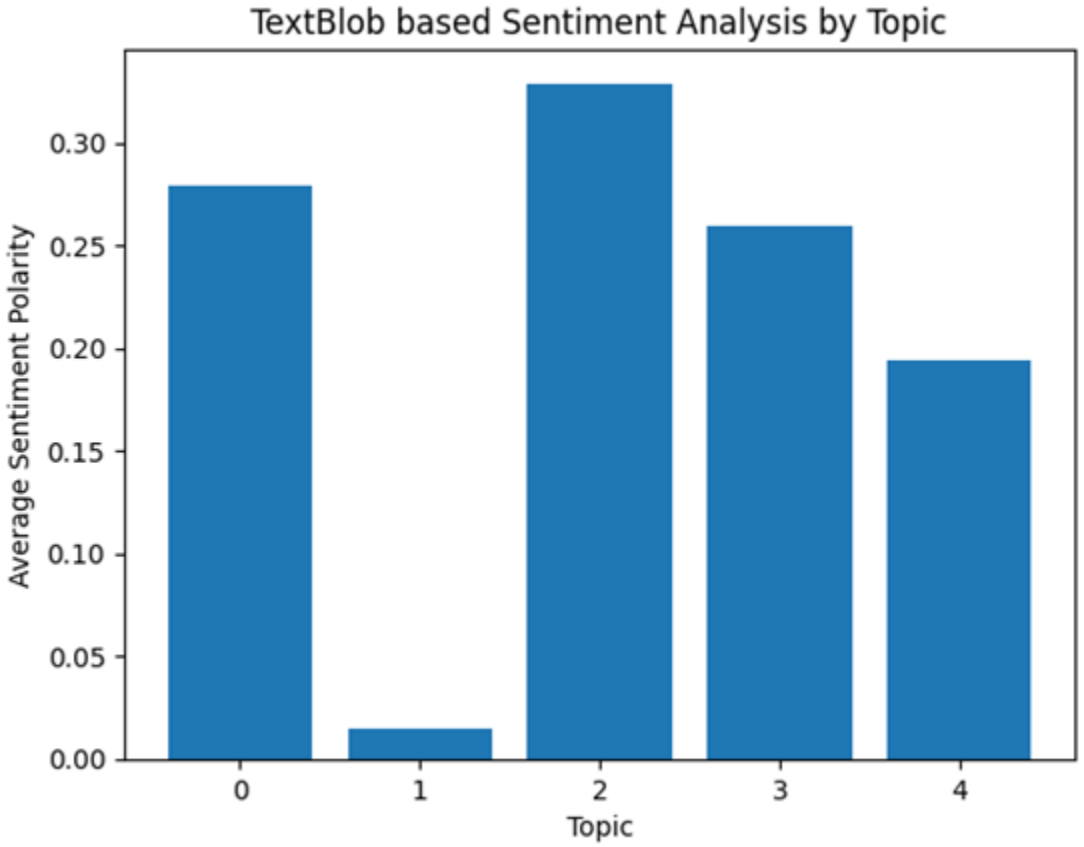


Rating 1

Sentiment Comparison (VADER and TextBlob)

Similar topic sentiment using 2 methods





Number of topics = 5

LDA (GENSIM)

Model Performance

Number of processed documents = 20,491

Total number of words in all reviews is 2,136,192

Total number of words in all reviews after processing is 1,992,542

Total number of unique words in all reviews after processing is 77,149

Number of Topics	Perplexity Score	Coherence Score
5	-8.036570807887816	0.3449936728248204
50	-10.689358425894168	0.33776505076046653
100	-12.306198254812768	0.3524896845681038
200	-16.311778551890004	0.3399224743615561

Dataset Details and Data Clean Up Methods

Sajal Debnath

Dataset: New York Airbnb Open Data – Kaggle

- Work on subset of the data through column selection
- Removal of missing values
- Pre-processing:
  - **Replacing URLs**: Links starting with "http" or "https" or "www" are replaced by "URL".
  - **Replacing Emojis**: Replace emojis by using a pre-defined dictionary containing emojis along with their meaning. (eg: ":)" to "EMOJIsmile")
  - **Removing Non-Alphabets**: Replacing characters except Digits and Alphabets with a space.
  - **Removing Consecutive letters**: 3 or more consecutive letters are replaced by 2 letters. (eg: "Heyyyy" to "Heyy")
  - **Removing Short Words**: Words with length less than 2 are removed.
  - **Removing Stopwords**: Stopwords are the English words which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. (eg: "the", "he", "have")
  - **Lemmatizing**: Lemmatization is the process of converting a word to its base form. (e.g: “Great” to “Good”)
  - **Lower Casing**: Each text is converted to lowercase.
- Tokenization of words

*** Please check Sajal Debnath’s GitHub Repository for all the details.

Results – Sajal Debnath – Scikit-Learn Library - Coherence Score

- I am using Scikit-learn library to implement LDA mode
- I am using Coherence score using cosine similarity to calculate the effectiveness of the model
- The coherence score before hyper tuning is 0.2495
- After hyper tuning the best score is 0.6699

```
Model Coherence Score: 0.2495
```

N_topics: 3

Learning_decay:

0.7

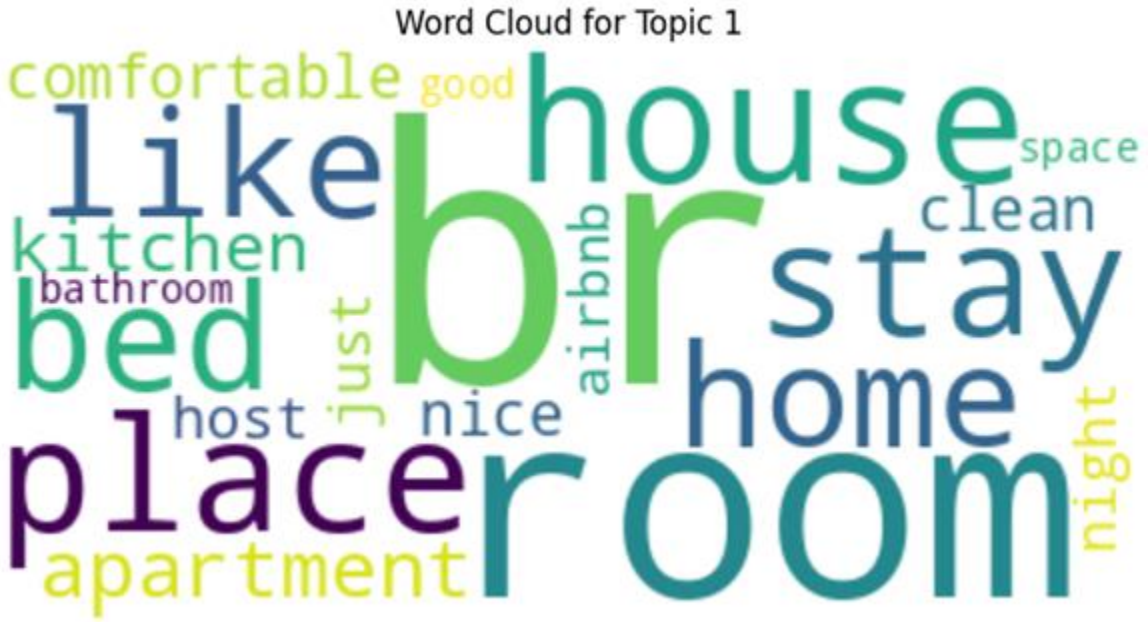
Learning_offset:

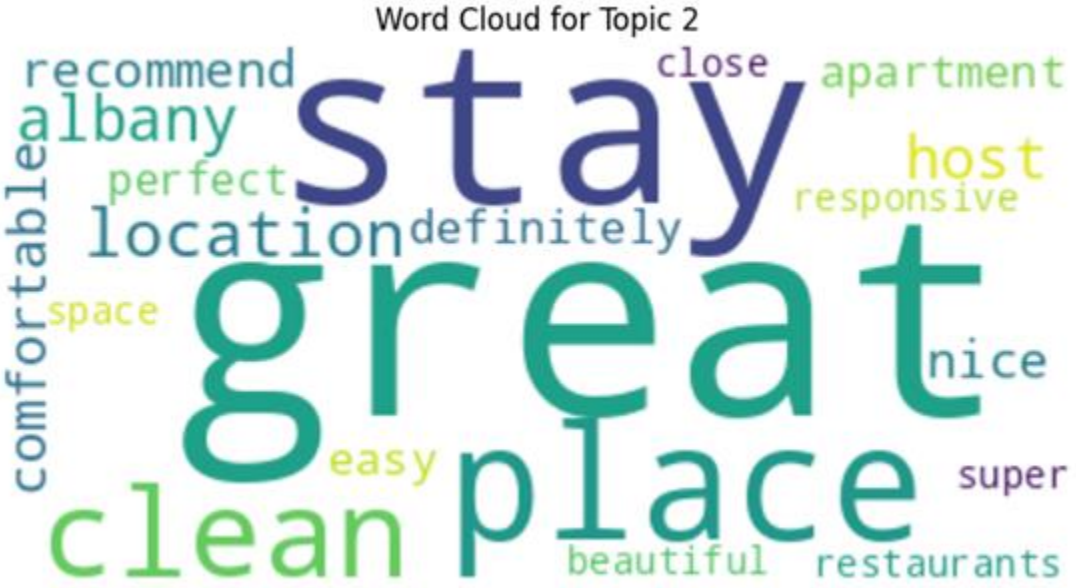
10

```
Tuning n_components:   0%|          | 0/3 [00:00<?, ?it/s]n_topics: 2, learning_decay: 0.5, learning_offset: 10, score: 0.6699
n_topics: 2, learning_decay: 0.5, learning_offset: 50, score: 0.6699
n_topics: 2, learning_decay: 0.5, learning_offset: 100, score: 0.6699
n_topics: 2, learning_decay: 0.7, learning_offset: 10, score: 0.6699
n_topics: 2, learning_decay: 0.7, learning_offset: 50, score: 0.6699
n_topics: 2, learning_decay: 0.7, learning_offset: 100, score: 0.6699
n_topics: 2, learning_decay: 0.9, learning_offset: 10, score: 0.6699
n_topics: 2, learning_decay: 0.9, learning_offset: 50, score: 0.6699
Tuning n_components:  33%|███       | 1/3 [12:08<24:17, 728.91s/it]n_topics: 2, learning_decay: 0.9, learning_offset: 100, score: 0.6699
n_topics: 3, learning_decay: 0.5, learning_offset: 10, score: 0.5779
n_topics: 3, learning_decay: 0.5, learning_offset: 50, score: 0.5779
n_topics: 3, learning_decay: 0.5, learning_offset: 100, score: 0.5779
n_topics: 3, learning_decay: 0.7, learning_offset: 10, score: 0.5779
n_topics: 3, learning_decay: 0.7, learning_offset: 50, score: 0.5779
n_topics: 3, learning_decay: 0.7, learning_offset: 100, score: 0.5779
n_topics: 3, learning_decay: 0.9, learning_offset: 10, score: 0.5779
n_topics: 3, learning_decay: 0.9, learning_offset: 50, score: 0.5779
Tuning n_components:  67%|██████    | 2/3 [26:52<13:39, 819.90s/it]n_topics: 3, learning_decay: 0.9, learning_offset: 100, score: 0.5779
n_topics: 5, learning_decay: 0.5, learning_offset: 10, score: 0.5142
n_topics: 5, learning_decay: 0.5, learning_offset: 50, score: 0.5142
n_topics: 5, learning_decay: 0.5, learning_offset: 100, score: 0.5142
n_topics: 5, learning_decay: 0.7, learning_offset: 10, score: 0.5142
n_topics: 5, learning_decay: 0.7, learning_offset: 50, score: 0.5142
n_topics: 5, learning_decay: 0.7, learning_offset: 100, score: 0.5142
n_topics: 5, learning_decay: 0.9, learning_offset: 10, score: 0.5142
n_topics: 5, learning_decay: 0.9, learning_offset: 50, score: 0.5142
Tuning n_components: 100%|██████████| 3/3 [40:56<00:00, 818.78s/it]n_topics: 5, learning_decay: 0.9, learning_offset: 100, score: 0.5142

Best Parameters:
n_topics: 2, learning_decay: 0.5, learning_offset: 10
Best Score: 0.6699

Topics in Best Model:
Topic 1: kitchen apartment home bed like stay place house room br
Topic 2: recommend nice comfortable host albany location clean place stay great
```

Results - 1

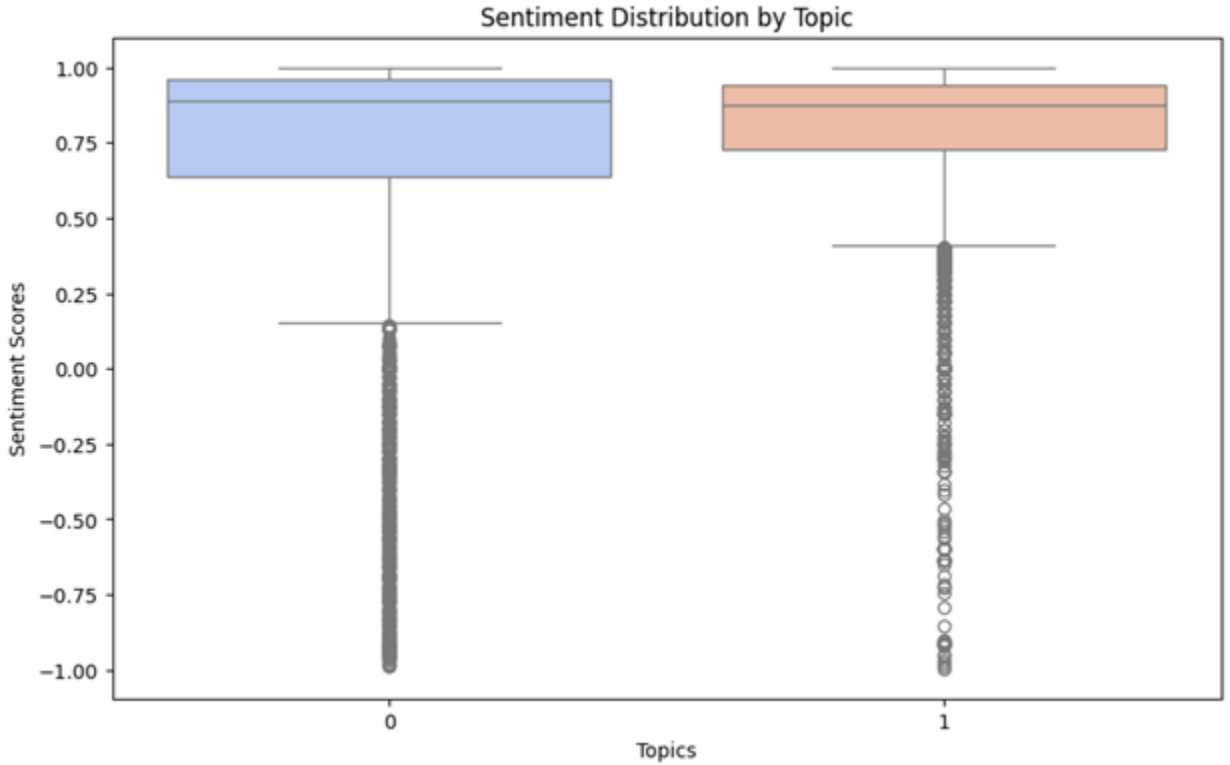
Wordcloud

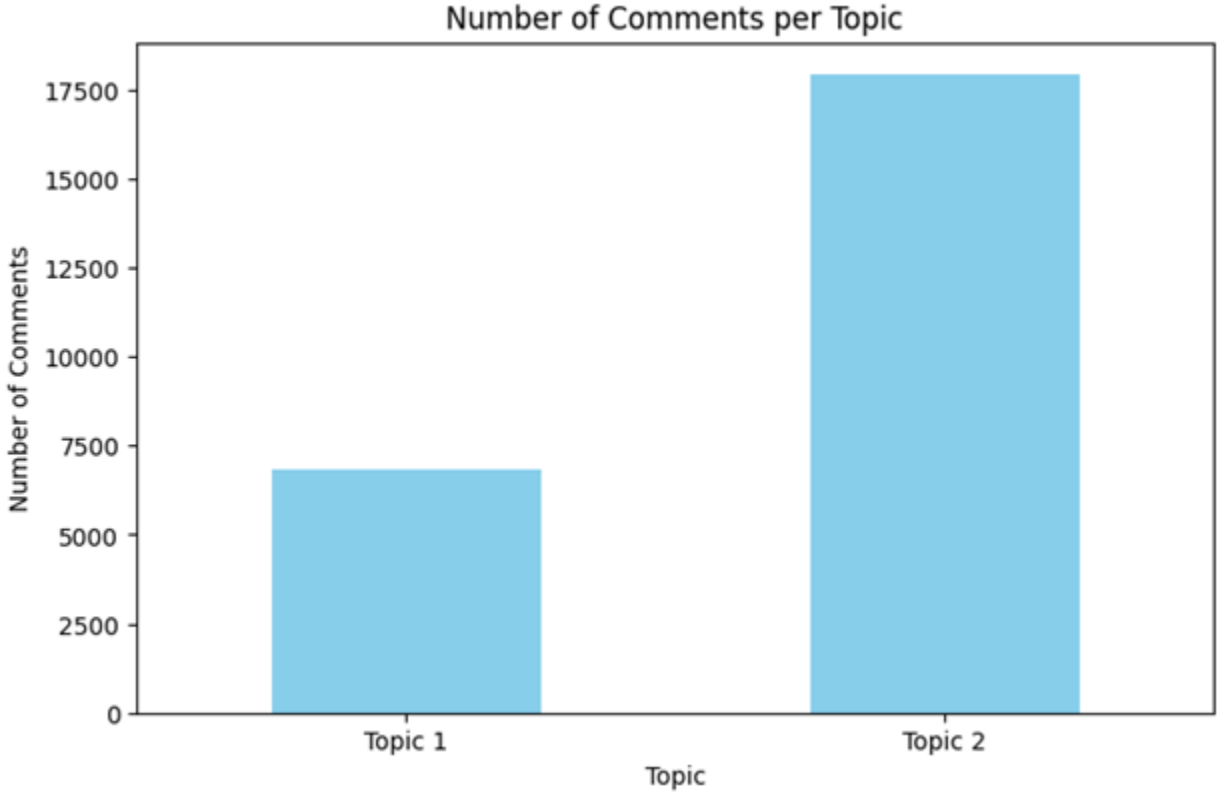




Results – 2

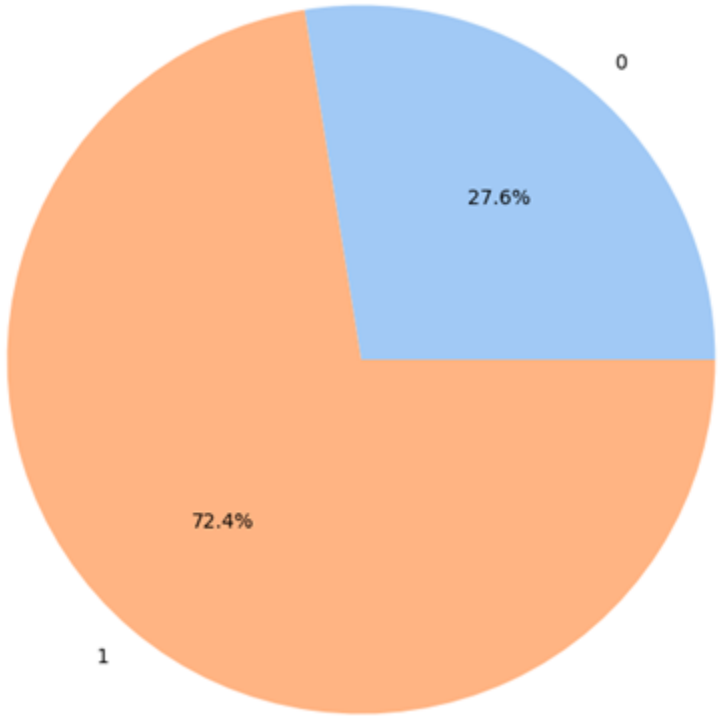
Sentiment Distribution, No. of Comments/topic

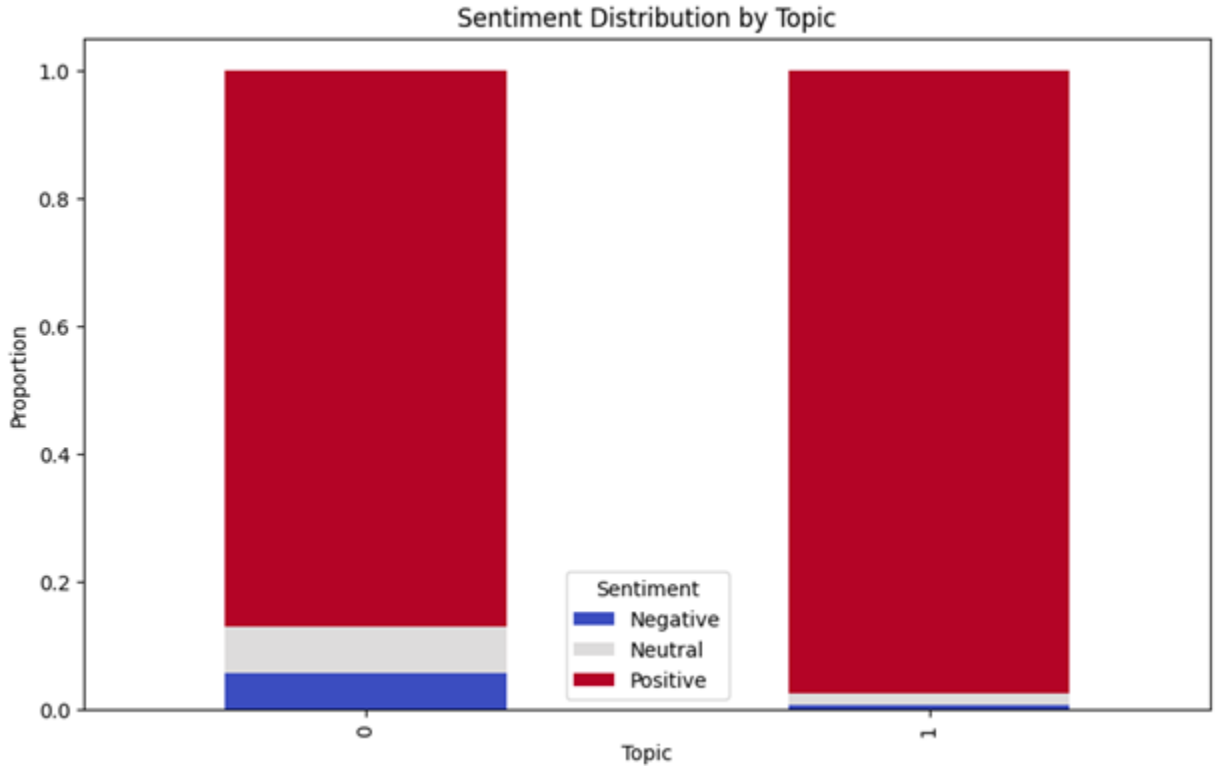




Results - 3

Comments/topic , Sentiment distribution/topic

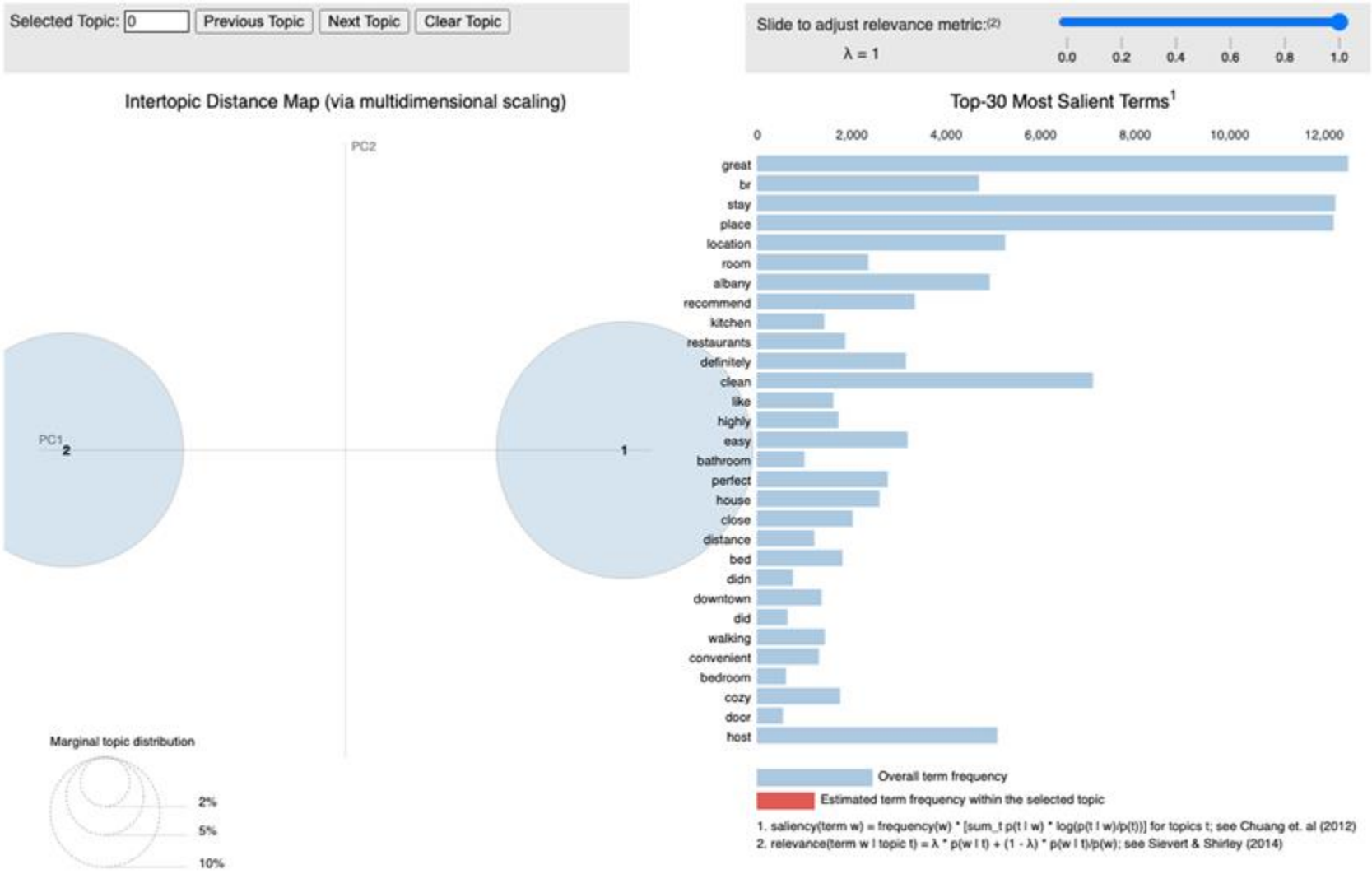




Proportion of Comments per Topic

Results – 4

Interactive Visualization with pyLDAvis



Contributions

Sajal Debnath

- Data cleanup and acquisition - 50 / 100
- ML Model Selection/ Training - 50/ 100
- Hyper parameter tuning - 50 /100
- Metrics - 50/ 100
- Presentation - 50 / 100
- Documentation - 50 / 100

Krishanu Chattopadhyay

- Data cleanup and acquisition - 50 / 100
- ML Model Selection/ Training - 50 / 100
- Hyper parameter tuning - 50 /100
- Metrics - 50 / 100
- Presentation - 50/ 100
- Documentation - 50 / 100

Conclusion

- In conclusion we can say that while LDA is a very powerful algorithm, there are other models we need to explore. For example, pLSA model and others. Also, we need to continue working on the datasets and the model to make it more useful.
- We need both time and effort to fine tune the models and adopt it as per the datasets.
- For the Airbnb dataset, the sentiment is overwhelmingly positive, this is due to the nature of the reviews. We need to test the model with other datasets with actual rating and compare them with the generated ratings to find out the accuracy of the model.
- For generating the LDA model with TripAdvisor dataset, the model was trained in 1 pass. The library gave a warning that the model was not converging, and more iteration needs to be performed. However, with the hardware provided in Google Colab runtime, this was not possible.

Thank You