Title: Clothing Recommendation Deep Learning Model

1. Main objective of the analysis that also specifies whether your model will be focused on a specific type of Deep Learning or Reinforcement Learning algorithm and the benefits that your analysis brings to the business or stakeholders of this data.

- The objective of the analysis is to use bidirectional long short-term memory (bi-LSTM) algorithm for clothing item recommendation after combining both numerical and text features from reviewers as one textual feature. Bi-LSTM is a Neural Network architecture by making use of information from both directions of the sequence. The model will benefit both customers and sellers in making good recommendation for next clothing shopping.

2. Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.

- The data is called Women's E-Commerce Clothing Reviews, downloadable via https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews.
- It is real commercial review data provided by customers and hence the data has been anonymized.
- The data is in CSV file format and contains 10 data columns as follows:
  - Clothing ID: Integer Categorical variable that refers to the specific piece being reviewed.
  - Age: Positive Integer variable of the reviewers age.
  - Title: String variable for the title of the review.
  - Review Text: String variable for the review body.
  - Rating: Positive Ordinal Integer variable for the product score granted by the customer from 1 Worst, to 5 Best.
  - Recommended IND: Binary variable stating where the customer recommends the product where 1 is recommended, 0 is not recommended.
  - Positive Feedback Count: Positive Integer documenting the number of other customers who found this review positive.
  - Division Name: Categorical name of the product high level division - General, General Petitie and Initmates, Tops, Trend.
  - Department Name: Categorical name of the product department name – Bottoms, Dresses, Intimate, Jackets,
  - Class Name: Categorical name of the product class name e.g. Blouses, Dresses, Jeans and etc.
- Bi-LSTM algorithm was used to develop clothing item recommendation model based on complete data by combining both numerical and text features from reviewers as one

textual feature, ignoring 'Clothing ID' and 'Positive Feedback Count '. The target feature was the binary 'Recommended IND'.

- The final textual review as the input feature by combining both numerical and text features was done as follows [5]:

> This item comes from the {Department Name} and {Division Name}, and is classified under {Class Name}. I am {Age} years old. I rate this item {Rating} out of 5 stars. {Title}. {Review Text}

- A stratified train-test split on data without any missing values with 80% training and 20% testing was used due to class imbalance.
- Only lower-case conversion and removing of leading, trailing and multiple spaces from the text review were performed. Removing of stop words or punctuations from the final textual review feature were not performed to ensure its context.
- A tokenizer to tokenize all the words of the combined text and create sequences of tokenized words was carried out. Any Out of Vocab token i.e. unknown words was replaced by <OOV> instead of throwing them away since they contained information for the model to learn.
- Sequence padding to maximum number of words of 173 was applied to all reviews.
- In brief, the Bi-LSTM model consists of embeddings, bidirectional with LSTM followed by two dense layers [6].
- 80% training data generated from the train-test split was divided further into 10% for cross validation and 90% to train the Bi-LSTM model with shuffle before each epoch.

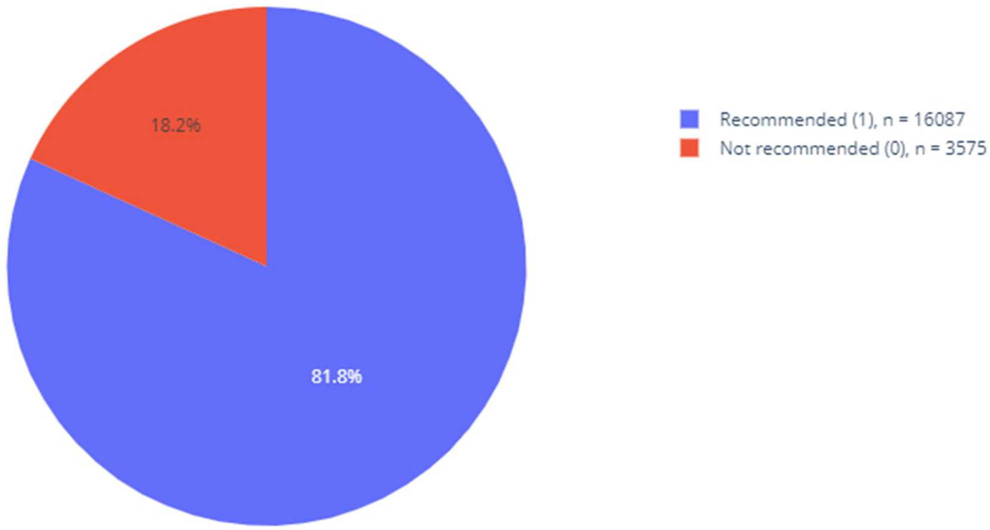
3. Brief summary of data exploration and actions taken for data cleaning or feature engineering.

- The data contains 23486 entries in total with 3824 entries with at least one missing values.
- Following are the last five entries:

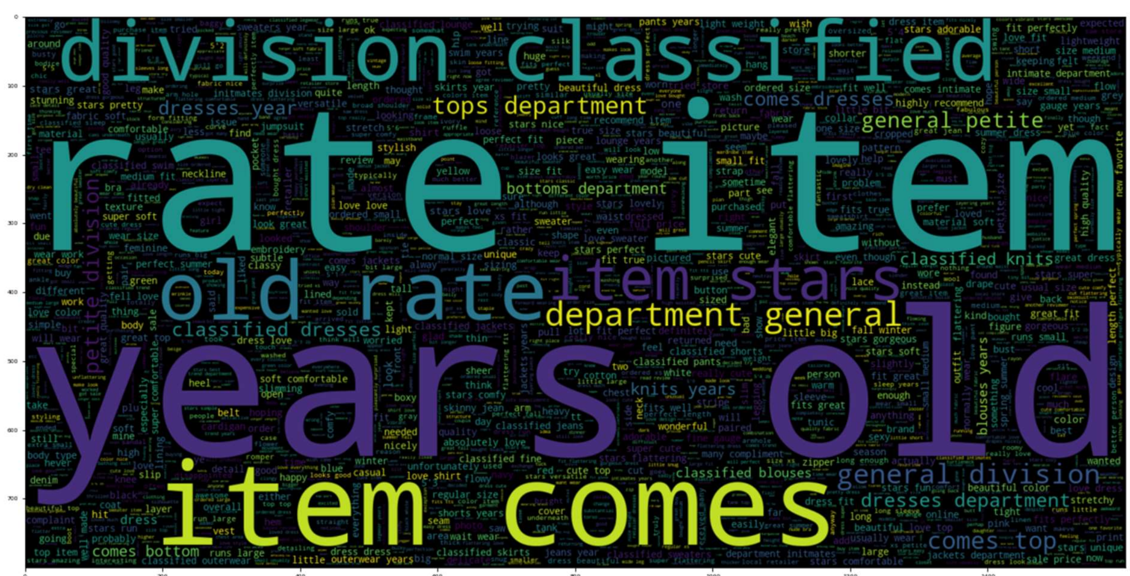
	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
23481	1104	34	Great dress for many occasions	I was very happy to snag this dress at such a great price! it's very easy to slip on and has a very flattering cut and color combo.	5	1	0	General Petite	Dresses	Dresses
23482	862	48	Wish it was made of cotton	It reminds me of maternity clothes. soft, stretchy, shiny material. cut is flattering and drapes nicely. i only found one button to close front... looked awkward. nice long sleeves.\nnot for me but maybe for others. just ok.	3	1	0	General Petite	Tops	Knits
23483	1104	31	Cute, but see through	This fit well, but the top was very see through. this never would have worked for me. i'm glad i was able to try it on in the store and didn't order it online. with different fabric, it would have been great.	3	0	1	General Petite	Dresses	Dresses
23484	1084	28	Very cute dress, perfect for summer parties and we	I bought this dress for a wedding i have this summer, and it's so cute. unfortunately the fit isn't perfect. the medium fits my waist perfectly, but was way too long and too big in the bust and shoulders. if i wanted to spend the money, i could get it tailored, but i just felt like it might not be worth it. side note - this dress was delivered to me with a nordstrom tag on it and i found it much cheaper there after looking!	3	1	2	General	Dresses	Dresses
23485	1104	52	Please make more like this one!	This dress in a lovely platinum is feminine and fits perfectly, easy to wear and comfy, too! highly recommend!	5	1	22	General Petite	Dresses	Dresses

- Total number of entries without any missing values of 19662 was used for analysis.
- Both Clothing ID and 'Positive Feedback Count' were not considered for analysis since former adds no value and the latter is related to other customers.

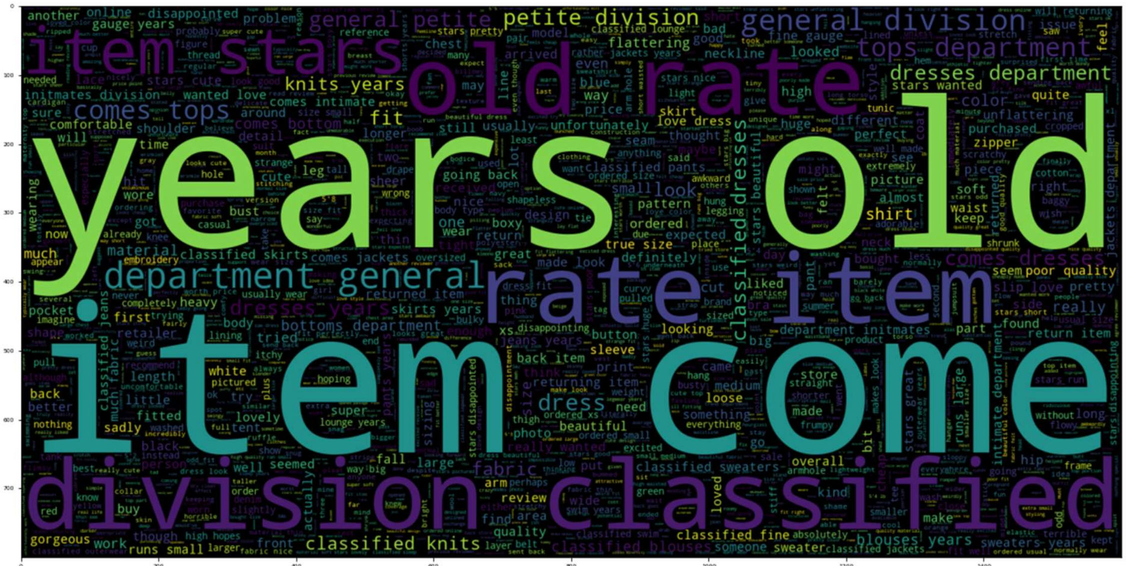
- Data has no duplicates.
- The data is very imbalanced.



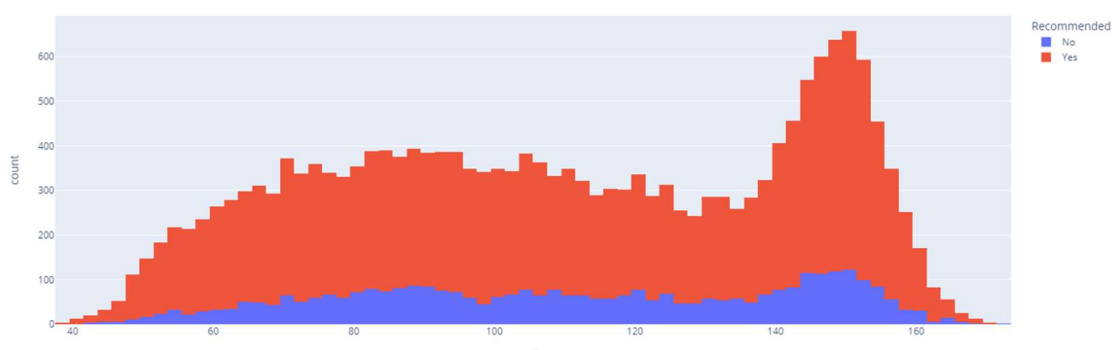
- The following is the word cloud based on the combined text review when the reviewers recommended the clothing item. The most seen words e.g. rate, item and etc are present in all reviews due to feature engineering by combining the text and numerical columns. Most recommended clothing items seem to be from division name of 'general' and department name of 'tops'.



- The following is the word cloud based on the combined text review when the reviewers did not recommend the clothing item. The most seen words e.g. rate, item and etc are present in all reviews due to feature engineering by combining the text and numerical columns. Most recommended clothing items seem to be from division name of ‘general’, department name of ‘dresses’ and class name of ‘knits’ or ‘dresses’.

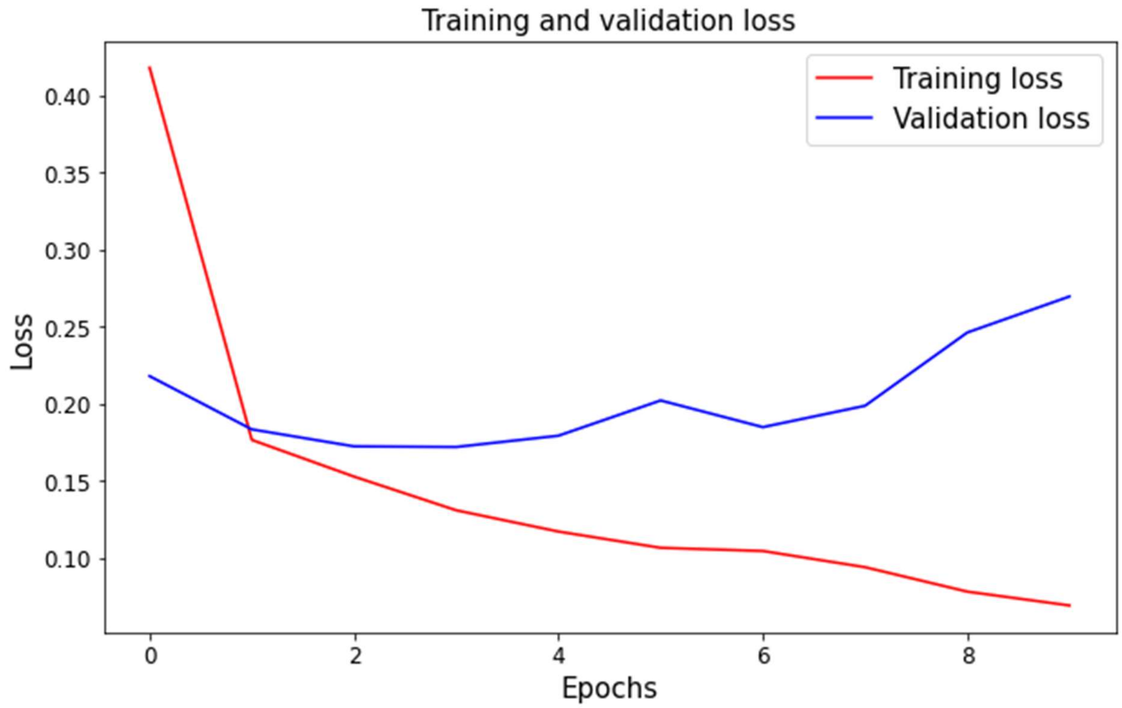


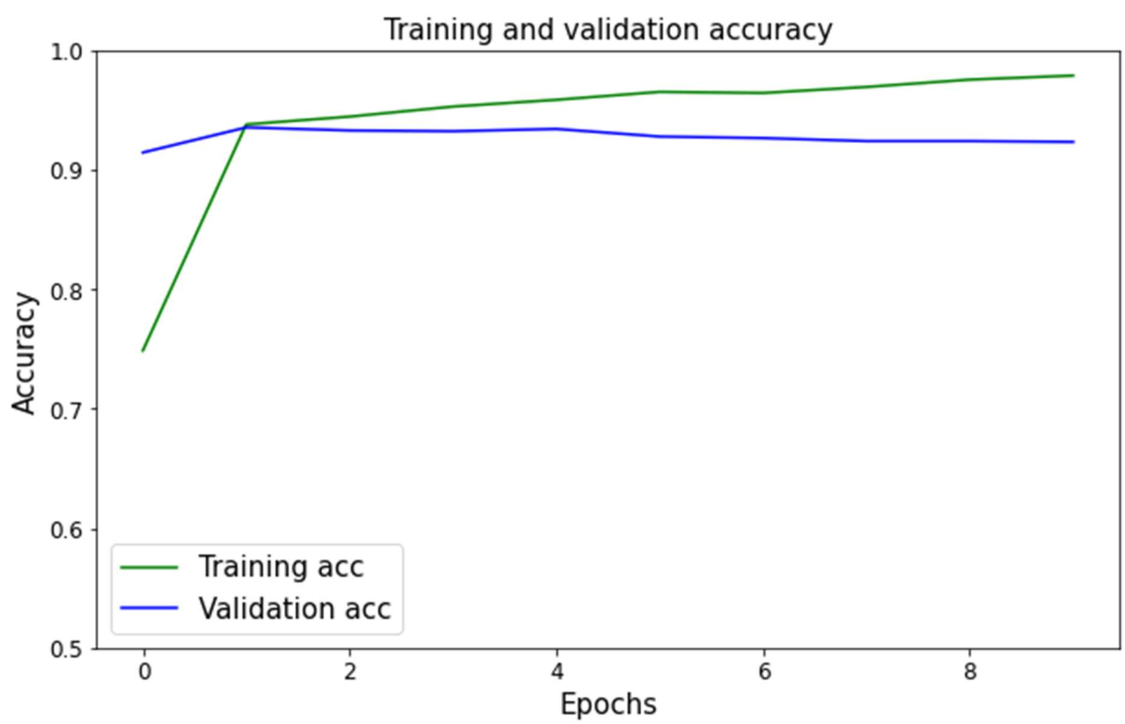
- The following is the histogram showing the distribution of number of words in a text by recommended groups. Most of the combined text reviews are within 140 words.



4. Summary of training at least three variations of the Deep Learning model you selected. For example, you can use different clustering techniques or different hyperparameters.

   i. *Setting epochs=10 and batch_size=32(default) but without using early stop*



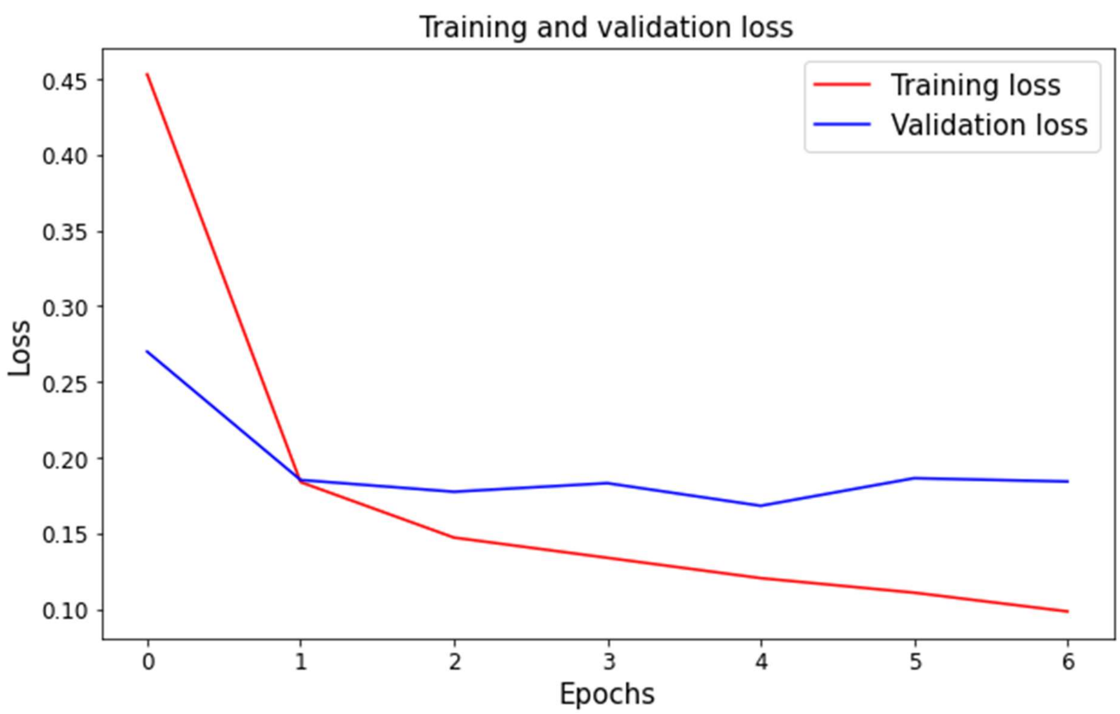


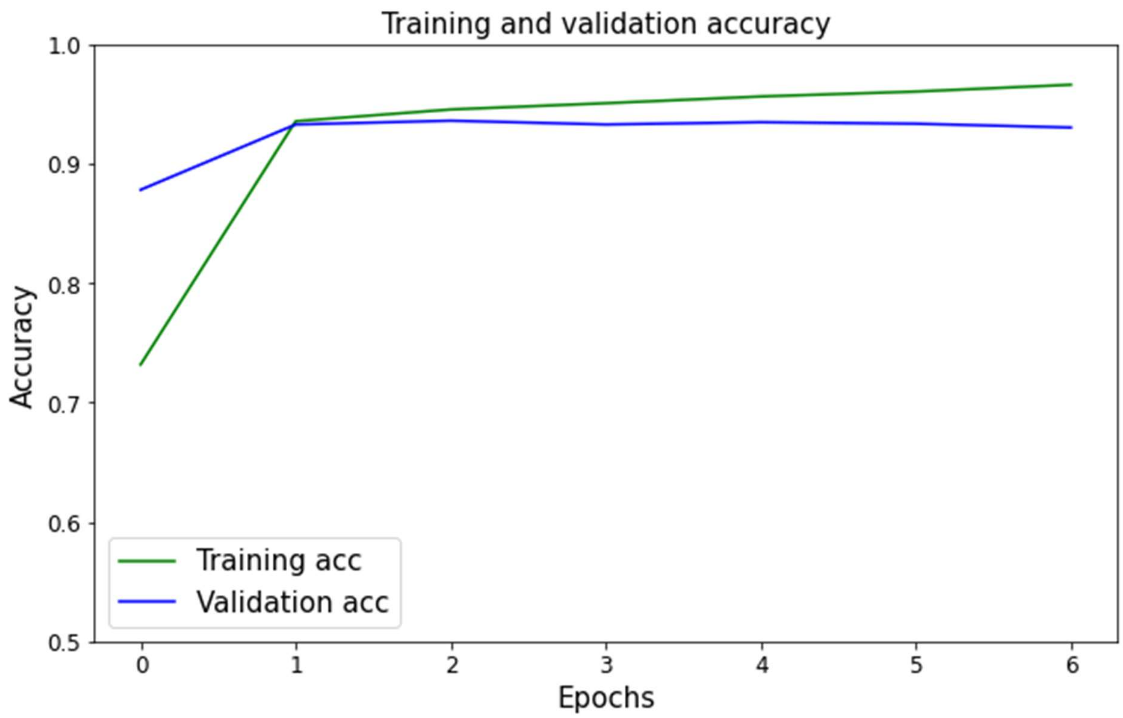
```
Accuracy on testing set: 0.9178743961352657
Precision on testing set: 0.9728846782097353, meaning for every
review that's recommended, there is about 97% chance of being
correct.
Recall on testing set: 0.925419515226849, the model can actually
identify about 93% of all reviews being recommended, meaning the
rest of 7% are being mis-classified as not recommended.
F1 score on testing set: 0.9485586876891224

Time: 2242.622913837433
```

Training stopped at 7th epoch i.e. the model will start to get overfitted from 8th epoch onwards. Therefore, the optimal number of epochs to train most dataset is 7. As the number of epochs increases beyond 7, training set loss decreases and approaches zero. Whereas, validation loss increases indicating the overfitting of the model on training data.

ii. *Setting epochs=10 and batch_size=32 (default) with early stop which stops when the validation loss no longer improves*

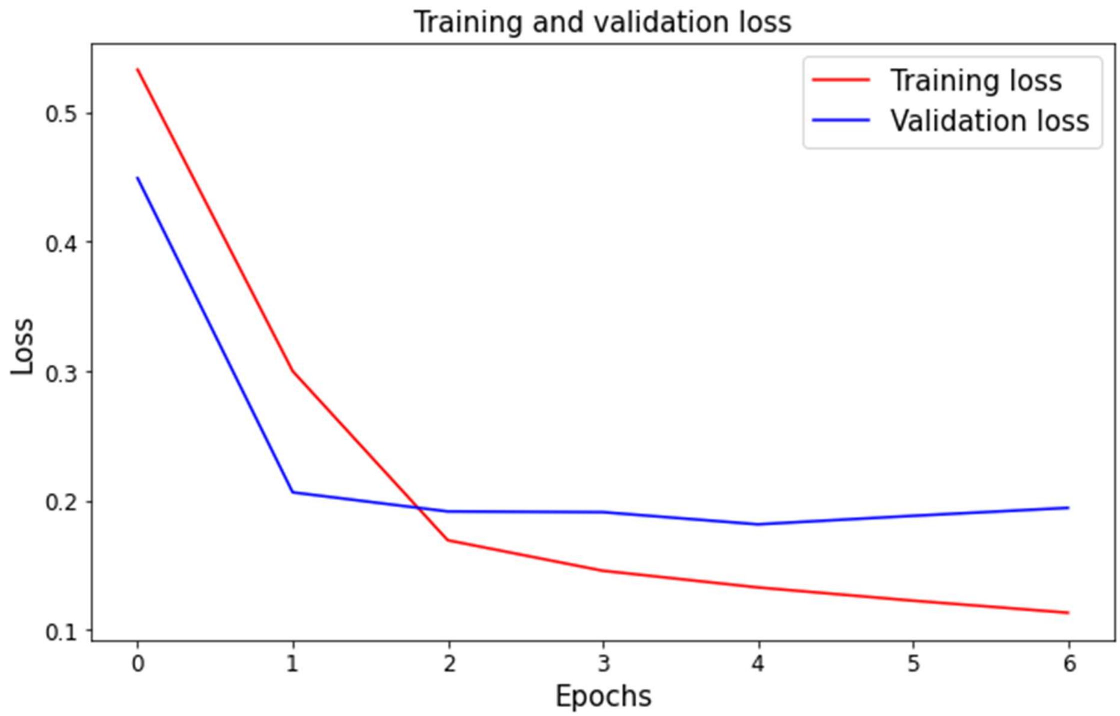


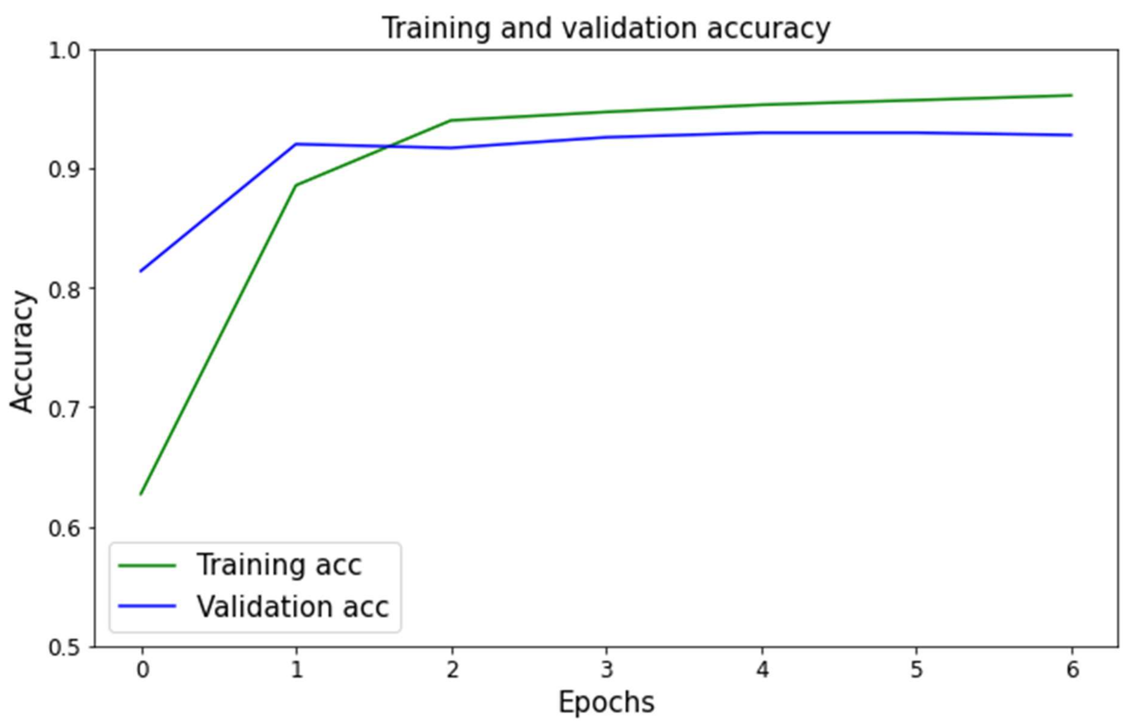



```
Accuracy on testing set: 0.8733790999237223
Precision on testing set: 0.9963503649635036, meaning for every
review that's recommended, the chance of being correct is nearly as
high as 100%.
Recall on testing set: 0.8483530142945929, the model can actually
identify about 85% of all reviews being recommended, meaning the
rest of 15% are being mis-classified as not recommended.
F1 score on testing set: 0.916414904330312

Time: 1510.0934765338898
```

iii. *Setting epochs=10 and batch_size=64 with early stop which stops when the validation loss no longer improves*






```
Accuracy on testing set: 0.9125349605898805
Precision on testing set: 0.992798353909465, meaning for every review
that’s recommended, the chance of being correct is about 99%.
Recall on testing set: 0.8996270975761342, the model can identify about
90% of all reviews being recommended, meaning the rest of 10% are being
mis-classified as not recommended.
```

```
F1 score on testing set: 0.9439191392239973

Time: 1167.2371129989624
```

Table of Performance Metrics of *i*, *ii and iii*:

Performance metrics	***i.*** **Without stopping rule, epochs=10, batch size=32**	***ii.*** **With stopping rule, epochs=10, batch size=32**	***iii.*** **With stopping rule, epochs=10, batch size=64**	**Recommend**
Accuracy	0.917874396	0.8733791	0.912534961	X
Precision	0.972884678	0.996350365	0.992798354	ii ≈ iii
Recall	0.925419515	0.848353014	0.899627098	i
F1 score	0.948558688	0.916414904	0.943919139	i ≈ iii
Time taken (seconds)	2242.622914	1510.093477	1167.237113	iii
Time taken (minutes)	37.37704856	25.16822461	19.45395188	iii

5. A paragraph explaining which of your Deep Learning models you recommend as a final model that best fits your needs in terms of accuracy or explainability.

- Looking at the above table on performance metrics, the last deep learning model i.e. with stopping rule, epochs=10, batch size=64 is recommended as compared to the other two. Accuracy is not an appropriate performance metric given the data imbalance. Precision is similar and better for both *ii* and *iii*. Recall score is best for *i* and as well as its F1 score however the latter is not so much different form *iii*. *iii* took the shortest time to complete the whole analysis time. Hence, considering both the analysis time taken and performance metrics, model *iii* is therefore recommended. As far as the analysis objective is concerned, the cost associated with false positive i.e. if the actual clothing item review that is not recommended has been predicted as recommended, is higher than the cost of false negative i.e. if the actual clothing item review was misclassified as not recommended. This is helpful to us from a consumer perspective as we might not miss out the recommended ones for our next shopping experience.

6. Summary Key Findings and Insights, which walks your reader through the main findings of your modeling exercise.

- Fashion recommendation model has a huge impact on the buying decisions of consumers and uncovers the new opportunities for fashion retailers by enabling them to provide customised recommendations to consumers.

Through building a bi-LSTM model by studying both the numerical data and text review from customers [5] as a text feature seems to give a considerable satisfactory result on clothing recommendation for this data set according to the performance metrics.

7. Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model or adding specific data features to achieve a better model.

- Study the impact of missing data
- Apply the resampling methods due to class imbalance
- Collect more data
- Tune other hyperparameters
- Explore other deep learning algorithms without having to turn numerical features to text and incorporate the previous points

References

1. https://www.coursera.org/learn/deep-learning-reinforcement-learning/home/welcome
2. https://www.coursera.org/learn/supervised-machine-learning-classification/home/welcome
3. https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews
4. http://digital-thinking.de/deep-learning-combining-numerical-and-text-features-in-deep-neural-networks/#comments
5. https://mccormickml.com/2021/06/29/combining-categorical-numerical-features-with-bert/#s3-bert-with-all-features
6. https://actsusanli.medium.com/fake-news-classification-with-lstm-or-logistic-regression-82a3527aaf13
7. https://www.coursera.org/learn/nlp-fake-news-detector/home/welcome
8. https://www.kaggle.com/code/hamishdickson/using-keras-oov-tokens/notebook
9. https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer
10. https://keras.io/examples/nlp/bidirectional_lstm_imdb/
11. https://keras.io/api/
12. https://keras.io/api/models/sequential/
13. https://keras.io/api/callbacks/early_stopping/
14. https://keras.io/api/models/model_training_apis/#compile-method
15. https://keras.io/api/models/model_training_apis/#fit-method
16. https://www.geeksforgeeks.org/choose-optimal-number-of-epochs-to-train-a-neural-network-in-keras/
17. https://stats.stackexchange.com/questions/245502/why-should-we-shuffle-data-while-training-a-neural-network