As part of the research to build the ML model, I found that a recommender system would be an optimal solution since we are predicting ‘liked probabilities’ through user-item rating matrix. I have considered the below three approaches to find the solution.

1) Collaborative Filtering
2) Matrix Factorization
3) Deep Neural Networks (DNN)

Some of the limitations of collaborative filtering and matrix factorization methods include the user-items matrix complex relationships and data sparsity limits the model implementation. The aforementioned limitations can be addressed by using DNN model to predict ‘liked’ probabilities. As part of the assignment, I have explored the dataset in order get an initial insight from the data. I considered user and item as the main features for the model development as the covariate ‘x’ feature has a high chance of misclassification impact on the prediction.

Users and items were one-hot encoded and fed into the DNN as an input layer and ‘liked’ variable as the output. Deep Neural Network model was built by extracting the latent features of users and items with the help of TensorFlow Embedding layers and then Dense layers with dropouts were concatenated and finally a Dense layer with a Sigmoid activation function was added. I have used Hyperparameters for model tuning, many loss functions and optimizers were tried with minimum validation loss as metric to built the model. Finally, ‘adam’ optimizer and BinaryCrossentropy for loss function were used to compile the model. The classification report and confusion matrix show an accuracy of 79%.

The following are the confusion matrix, classification report and predicted “liked” probabilities for users 1 through 5 for all the items from the test set.

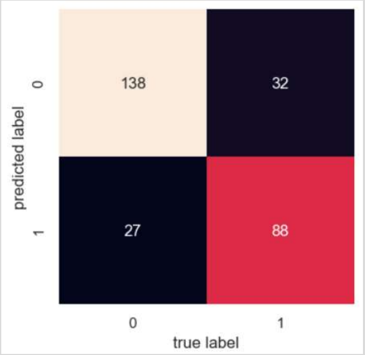
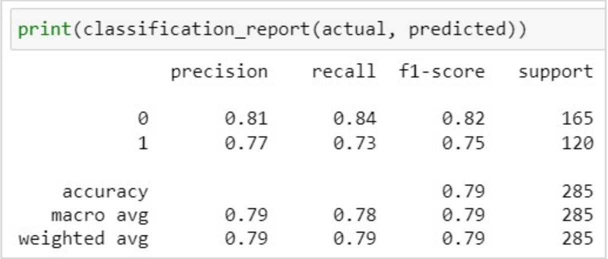


Figure 1: Confusion matrix



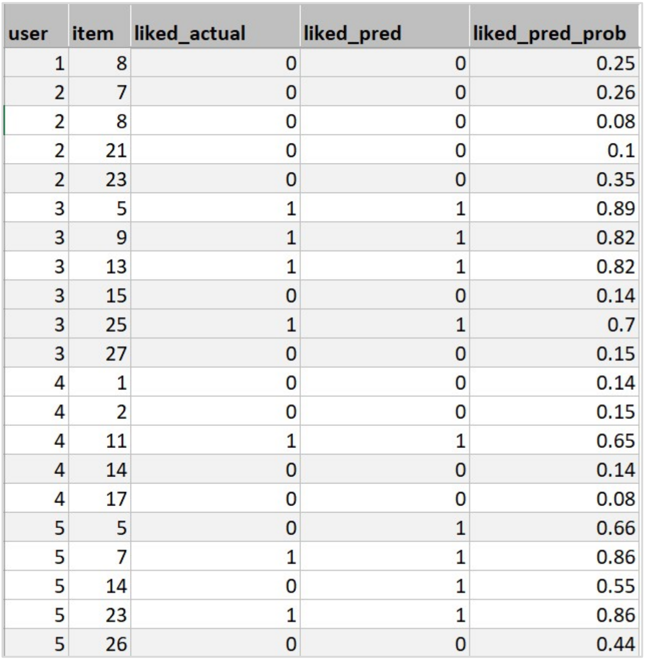
```
print(classification_report(actual, predicted))

              precision    recall  f1-score   support

           0       0.81      0.84      0.82       165
           1       0.77      0.73      0.75       120

    accuracy                           0.79       285
   macro avg       0.79      0.78      0.79       285
weighted avg       0.79      0.79      0.79       285
```

Figure 2: Classification Report



user	item	liked_actual	liked_pred	liked_pred_prob
1	8	0	0	0.25
2	7	0	0	0.26
2	8	0	0	0.08
2	21	0	0	0.1
2	23	0	0	0.35
3	5	1	1	0.89
3	9	1	1	0.82
3	13	1	1	0.82
3	15	0	0	0.14
3	25	1	1	0.7
3	27	0	0	0.15
4	1	0	0	0.14
4	2	0	0	0.15
4	11	1	1	0.65
4	14	0	0	0.14
4	17	0	0	0.08
5	5	0	1	0.66
5	7	1	1	0.86
5	14	0	1	0.55
5	23	1	1	0.86
5	26	0	0	0.44

Figure 3: Predicted ‘liked’ probabilities for first 5 users in the test set