Optimizing Promotional Offers For Starbucks Customers

SATYAJEET DAS, IIT Bombay

In this report, three datasets from Starbucks covering offer campaigns, customer transactions, and user profiles were analyzed to optimize promotional offers across different customer segments using various techniques like exploratory data analysis and pre-processing, machine learning techniques including random forest and XGBoost for identifying key patterns that influences customer behavior and offer interaction, statistical methods like PCA and t-SNE for dimentionality reduction and an experiment with auto-encoders to develop a multi-label prediction model for customer segmentation and most effective promotional offers

1 DATA DESCRIPTION

- **Portfolio Dataset**: Details on promotional offers along with features like reward, difficulty, duration etc
- **Profile Dataset**: Personal info of customers
- **Transcript Dataset**: Customer interactions with offers, a transactional dataset

2 EXPLORATORY DATA ANALYSIS

Data Cleaning: Removed entries with missing gender and income, which in terms helped get rid of outliers
Feature Engineering: Created new metrics: click rate, completion rate from the actions of offer received, viewed completed and transactional frequency and avg transactional Amount for better viewing users info with their actions **(fig 14)**
Statistical Analysis: Various distribution patterns in age, income, and offer interactions **(fig 1, fig 10, fig 11, fig 12, fig 13)**
One-Hot Encodings:For offers that were viewed and completed, the most frequently completed offers and the top three completed offers were calculated and created corresponding one-hot encodings. The former ones are used for a multi-class labeling classifier, while the latter are utilized for a multi-label auto-encoder model

3 DIMENSIONALITY REDUCTION WITH PCA

To reduce the complexity of the data and understand patterns, Principal Component Analysis (PCA) was employed initially. The goal was to reduce the number of variables to a manageable number while retaining significant variance

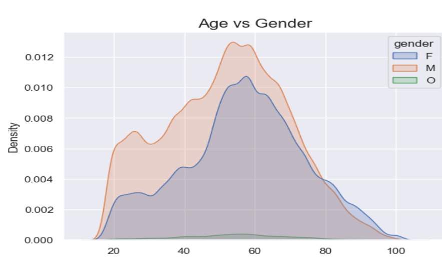

Fig. 1. Distribution of no. of users wrt to Age for different genders

Author's address: Satyajeet Das, satyajeetdas225@gmail.com, IIT Bombay.

4 ADVANCING TO T-SNE

While PCA provided a linear dimensionality reduction, it was insufficient in capturing variations, data points **(fig 14)** were getting forming blobbed reason being there could be some non-linear relationships or dataset being too homogeneous, evident from the clustered visualization of PCA results. To address this, t-Distributed Stochastic Neighbor Embedding (t-SNE) was utilized **(fig 15)**. It is good in identifying complex structures in high-dimensional data, ensuring that points close to each other in high dimensions remain close in the reduced space

5 EXPERIMENTATION WITH AUTO-ENCODERS

Autoencoders learn to encode the data into a lower-dim latent space and reconstruct it back to the original space thus capturing more nuances of dataset, Encoder part was given 7 dim informational features and it creates embeddings in 5 dim space, then the decoder part predicts top 3 offers that user completed so as to make it more flexible and corresponding target features were given to calculate the loss **(fig 16)** and backpropagate to train weights, this complete method being differentiable, so it has the potential to give decent results

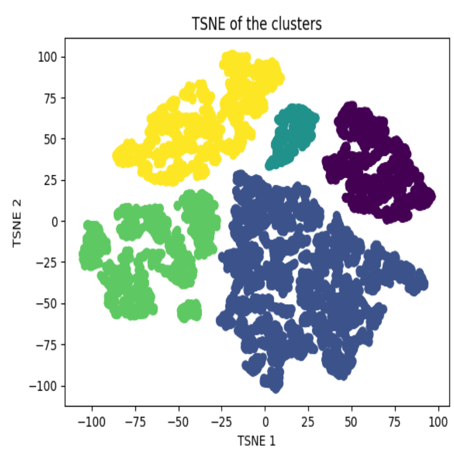

Fig. 2. Final Clusters using embeddings of autoencoders, lowered down to 2 dim using t-SNE and clustered using Kmeans

6 CLUSTERING AND CORRELATION ANALYSIS

With the insights gained from Autoencoders, Kmean clustering is applied to 5 segments, The objective is to minimize variances within cluster, This segmentation was further analyzed using a correlation matrix to understand how variables such as age, income, transaction frequency, and offer interaction metrics relate to each other.The elbow method **(fig 6)** was applied to determine an optimal number of clusters, Further analysis are given below

7 OBSERVATIONS AND INSIGHTS

- **Demographic Trends:** 57.1% of customers are male (average age 51) and 41.5% are female (average age 56), with a small percentage of missing data **(fig 10)**. Men generally had lower income levels than women **(fig 12)**
- **Offer Interaction:** Customers are interacting with offers extensively, with a 76% viewing rate and a 44% completion rate of offers. The analysis also showed distinct peaks in interaction every five days, corresponding to new releases **(fig 7)**
- Users are generally doing 4x more transactions rather than trying to complete offer, this is due to edible (FMCG) products where users don't care much about offers
- **Transactional Behavior:** Avg transaction frequency is left skewed, very higher number of users did 5 transactions and very few around 10 users did more than 30 on an average **(fig 21)**
- Income has positively correlation with avg transaction amount which is quite obvious but negative correlation with transaction frequency which is quite interesting **(fig 3)**
- Higher transaction frequency shows higher completion rate
- User with higher avg transaction amount tend to have a higher completion rate
- Younger users shows a higher transaction frequency, which may be due to factors such as increased spending possibly supported by parent or they just being too rebellious
- Females engage in more transactions than males
- Offer C and H **(fig 4)** were not chosen by any cluster, so those could be removed
- In fig 4 offers corresponding to highest count clusters are best best for those clusters

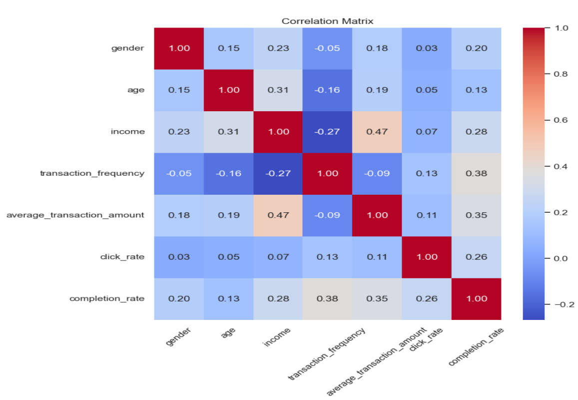

Fig. 3. Correlation matrix

8 PREDICTIVE MODELING

For predictive modeling, random forest classifier **(fig 18)** for input and target columns was initially used for its robustness in handling non-linear data, achieving a 59% accuracy rate, which improved to 61% after hyperparameter tuning which was done using Grid search to optimize the model resulting in higher accuracy. The model predicts customer offer completions based on demographics and past interactions

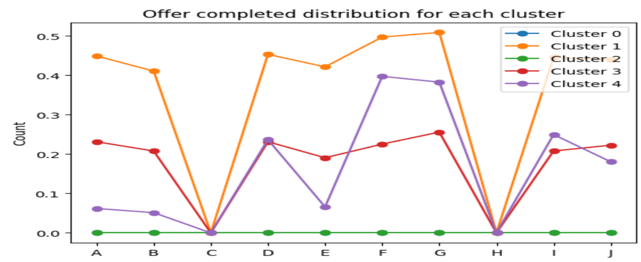

Fig. 4. Best offers with respect to customer clusters

9 EVALUATION AND ENHANCEMENT WITH XGBOOST

To further enhance the model, XGBoost was employed due to its capability to handle diverse data types and its efficiency in processing. To get a better understanding of how each offer is getting predicted F1 score is a good method to use, XGBoost also improved this score with an average F1 score of 69.4% across classes **(fig 20)**, offering a more balanced view between precision and recall, which is crucial for the multi-class classification task

10 CONCLUSION

In this study, A prediction model was developed for customer responses to offers With the help of Random Forest and XGBoost classifiers along with Autoencoder-based multi-label classification model to classify users in segments. The Random Forest model initial achieved accuracy of 59%, which was improved to 61% through hyperparameter tuning. The XGBoost model further increased the performance, achieving an average F1 score of 69.4% which is much higher than the base probability of 10% that a person will favor an offer

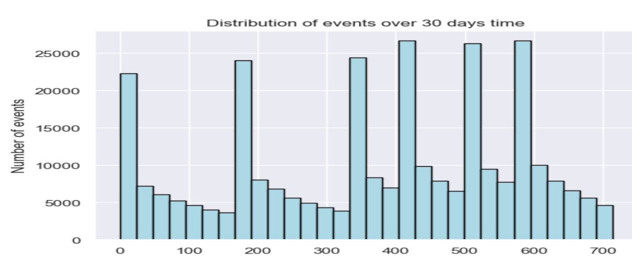

Fig. 5. Event Distribution over a period of 30 Days

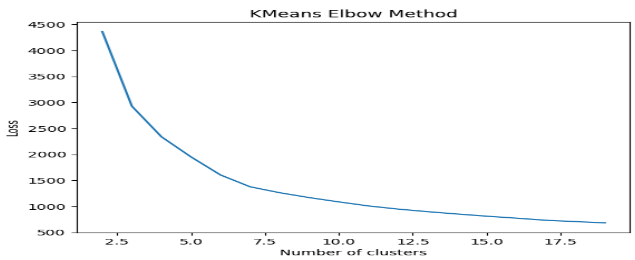

Fig. 6. Elbow Curve for finding optimal no. of clusters

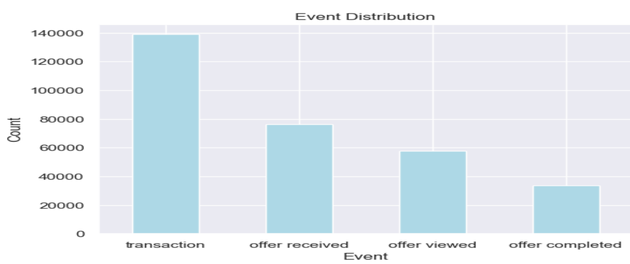

Fig. 7. Count of transactions, offers received viewed, and completed by users

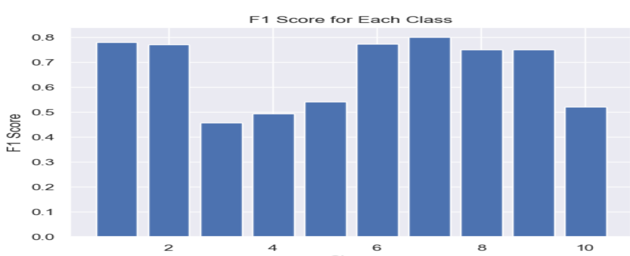

Fig. 8. F1 Score wrt to each offer predicted using random forest classifier

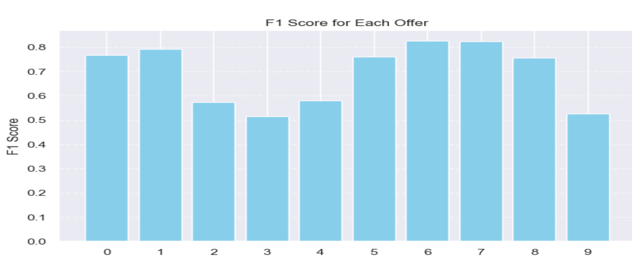

Fig. 9. F1 Score wrt to each offer predicted using XGBoost classifier

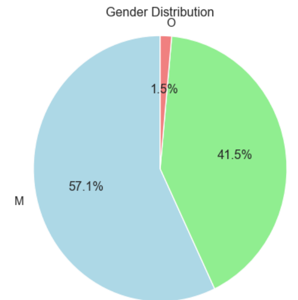

Fig. 10. Gender Distribution

Fig. 11. Distribution Of income

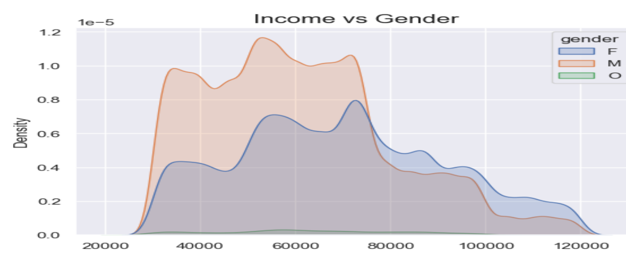

Fig. 12. Distribution of no. of users wrt to income for different genders

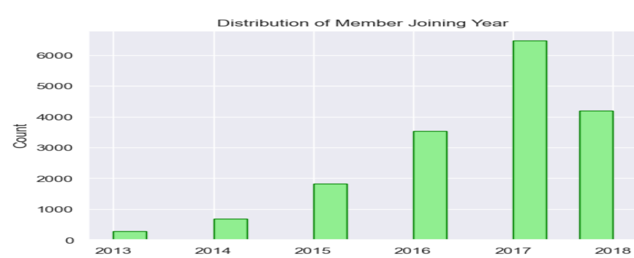

Fig. 13. No. of users wrt to joining Year

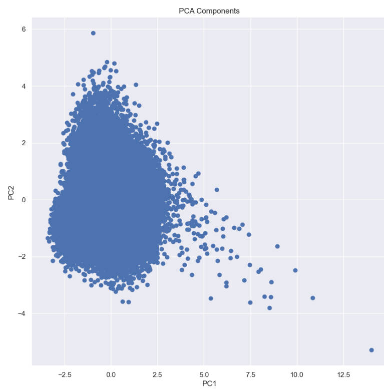

Fig. 14. Clusters using PCA and Kmeans

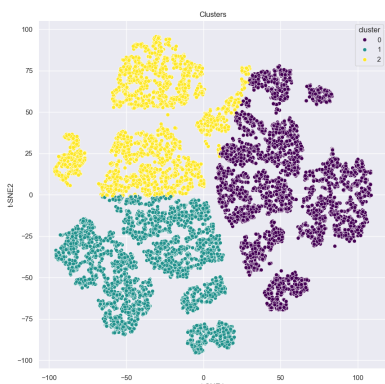

Fig. 15. Clusters just by using t-SNE and Kmeans

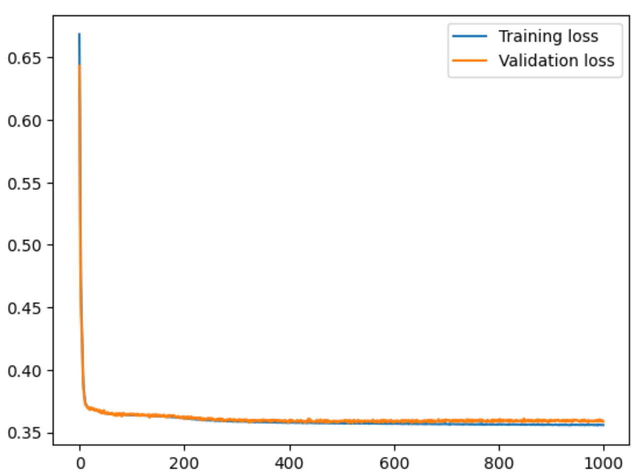

Fig. 16. Loss for Auto-encoders

	gender	age	income	transaction_frequency	average_transaction_amount	click_rate	completion_rate
cluster							
0	0.982716	57.603783	71699.331485	8.087885	16.955073	0.758242	0.565052
1	0.030106	55.366293	66048.701299	9.414109	20.259852	0.962525	0.561703
2	0.000000	49.879419	57483.699921	7.988610	8.901318	0.604219	0.341801

3 rows × 37 columns

Fig. 17. Avg user info for clusters made using t-SNE and Kmeans

```
X columns Index(['gender', 'age', 'income', 'transaction_frequency',
       'average_transaction_amount', 'click_rate', 'completion_rate',
       'offer_viewed_A', 'offer_viewed_B', 'offer_viewed_C', 'offer_viewed_D',
       'offer_viewed_E', 'offer_viewed_F', 'offer_viewed_G', 'offer_viewed_H',
       'offer_viewed_I', 'offer_viewed_J', 'cluster_0', 'cluster_1',
       'cluster_2', 'label_sum'],
      dtype='object')
Y columns Index(['most_completed_offer_A', 'most_completed_offer_B',
       'most_completed_offer_C', 'most_completed_offer_D',
       'most_completed_offer_E', 'most_completed_offer_F',
       'most_completed_offer_G', 'most_completed_offer_H',
       'most_completed_offer_I', 'most_completed_offer_J'],
      dtype='object')
```

Fig. 18. Input and Target features used to train the classifier

```
    plt.show()

✓ 0.2s

F1 score for class 1: 0.78
F1 score for class 2: 0.77
F1 score for class 3: 0.46
F1 score for class 4: 0.49
F1 score for class 5: 0.54
F1 score for class 6: 0.77
F1 score for class 7: 0.80
F1 score for class 8: 0.75
F1 score for class 9: 0.75
F1 score for class 10: 0.52
```

Fig. 19. F1 Score for each class, just to be sure that its not predicting only one class and still getting high score

```
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.74      0.77       431
           1       0.82      0.76      0.79       350
           2       0.80      0.45      0.57       137
           3       0.74      0.40      0.51       324
           4       0.77      0.47      0.58       431
           5       0.81      0.71      0.76       298
           6       0.88      0.78      0.83       271
           7       0.83      0.82      0.82        60
           8       0.72      0.80      0.76       430
           9       0.80      0.39      0.53       191

   micro avg       0.79      0.64      0.71      2923
   macro avg       0.80      0.63      0.69      2923
weighted avg       0.79      0.64      0.69      2923
 samples avg       0.63      0.64      0.63      2923


F1 Score for Each Class:
Class precision: 0.7882793837124967
Class recall: 0.6390694491960315
Class f1-score: 0.6943543590230439
Class support: 2923.0
```

Fig. 20. Final XGBoost Classification Report

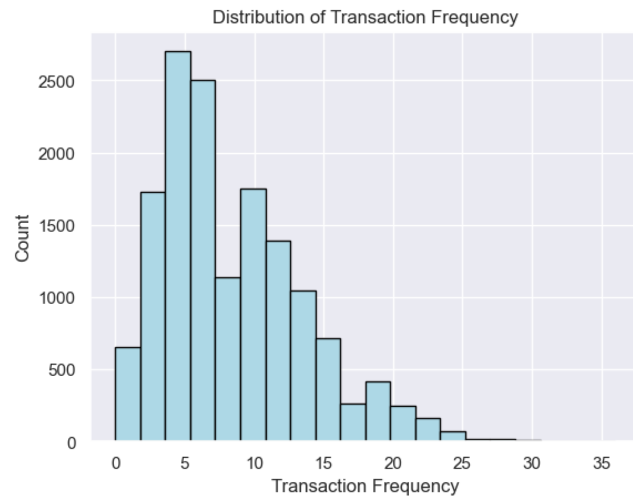

Fig. 21. Distribution of Avg transaction frequency