# JD Assignment for the post of Data Scientist

**Dr. Sandhya Jain**

26 September, 2024

# Goal

- Given the data at hand, The CRM team would like to send email campaigns with next best brand recommendation to customers who are most likely to repurchase in the next 3 months.

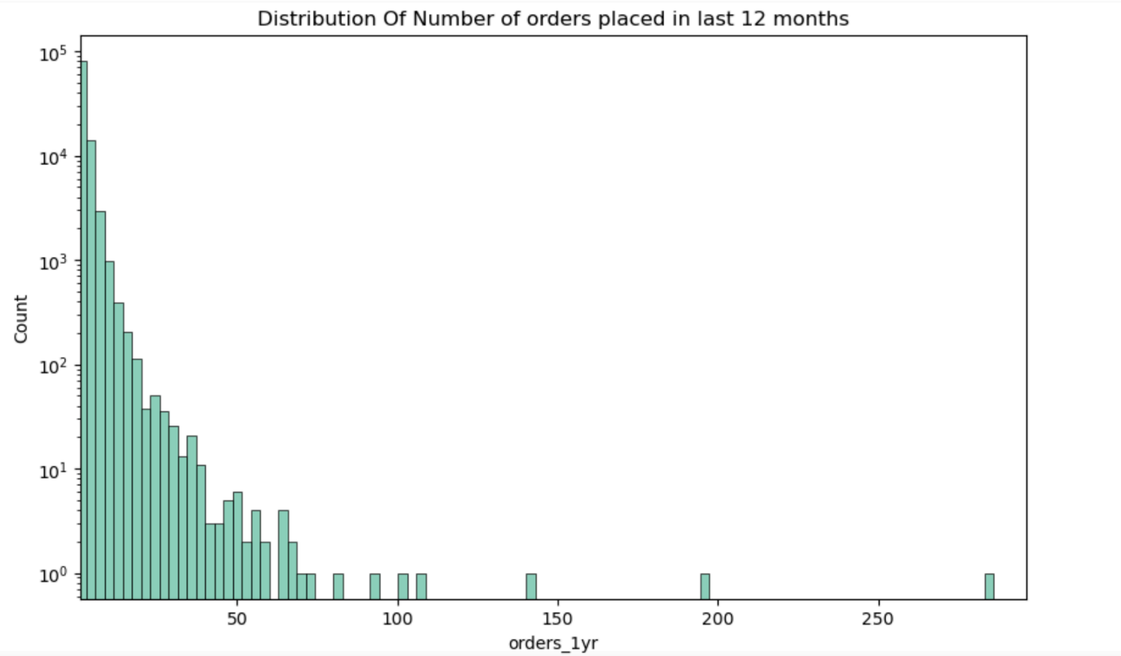
- Build a model based on the given dataset and identify which customers can be targeted. If each  email costs the company £2, what's the total misclassification cost?

# Approach

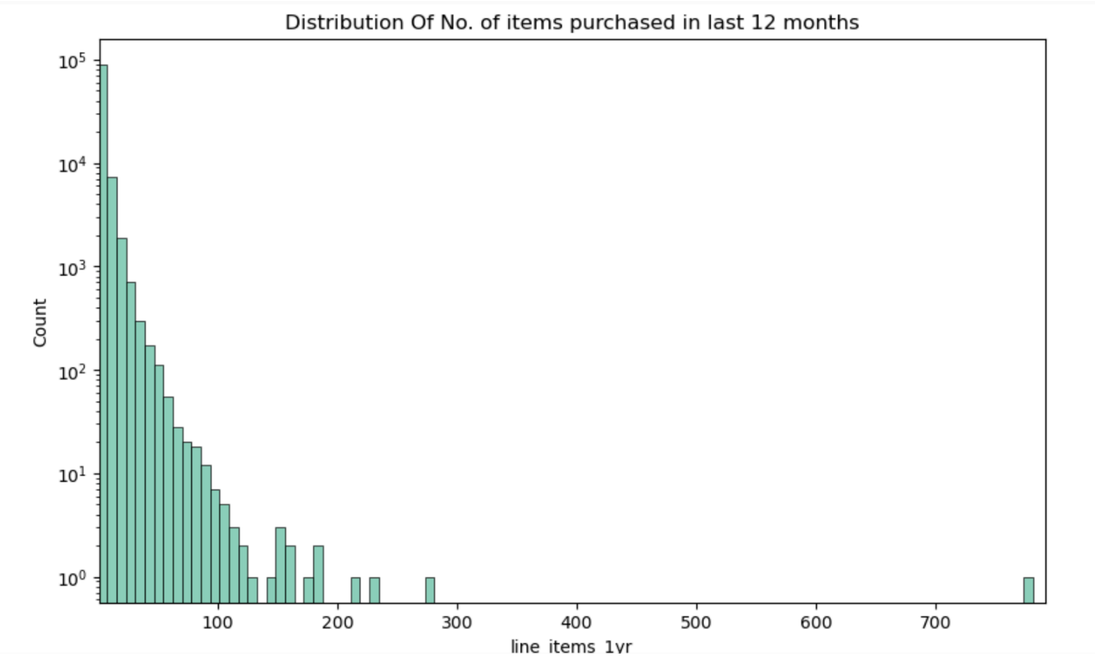
- Understand the data (Exploratory analysis!)
- Identify the modelling
- Make the model (prepare the model/preprocess the data/train the model/ evaluate the model)
- Results

# Exploratory analysis - I

- Several visualisations done to understand what we have at hand and what are the key variables in data.



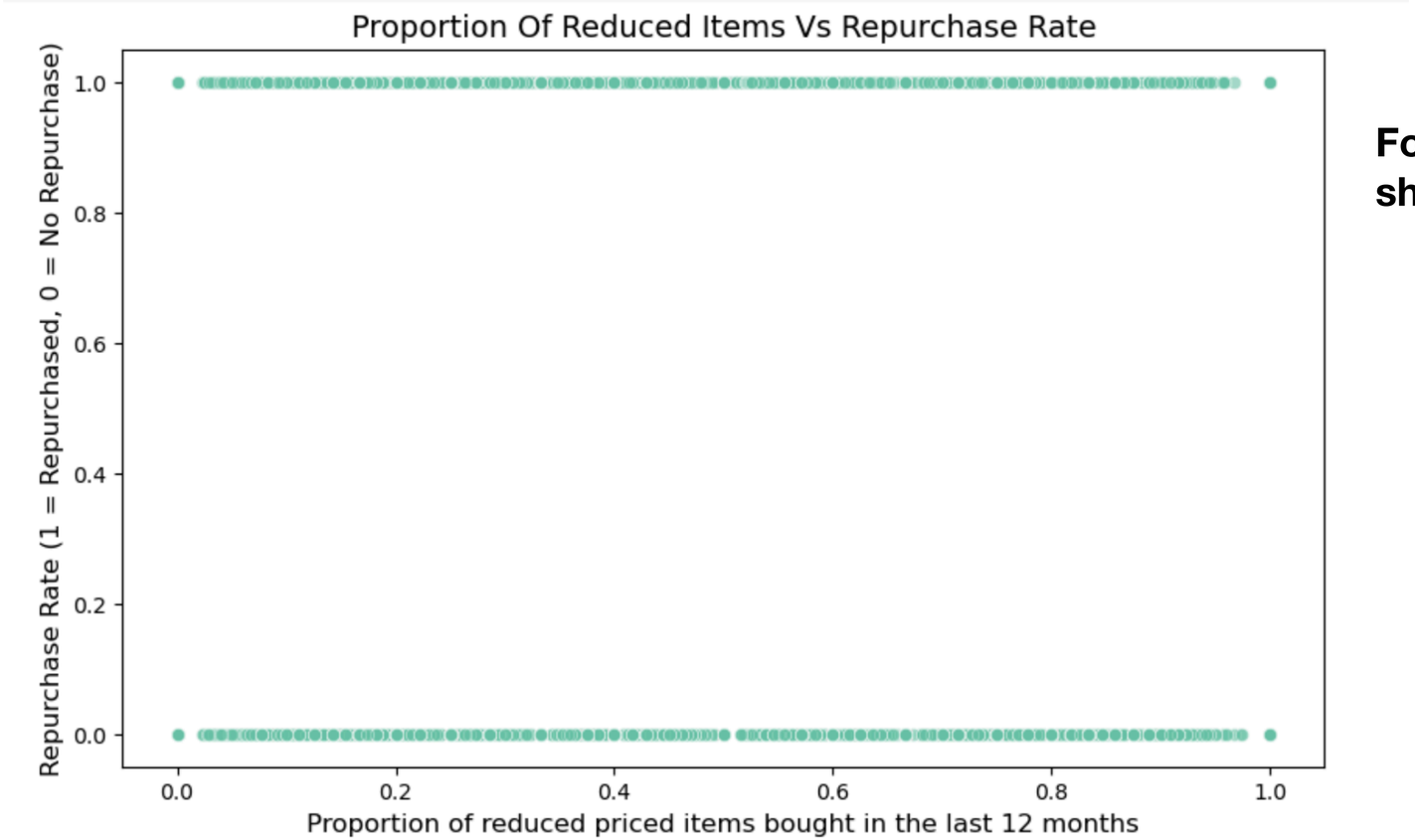

**The majority of customers placed 1 to 2 orders, but there are some outliers with significantly higher orders.**




**Most customers bought 1/2 items per order, with a few outliers with significant high number of items in the order.**

# Exploratory analysis - II

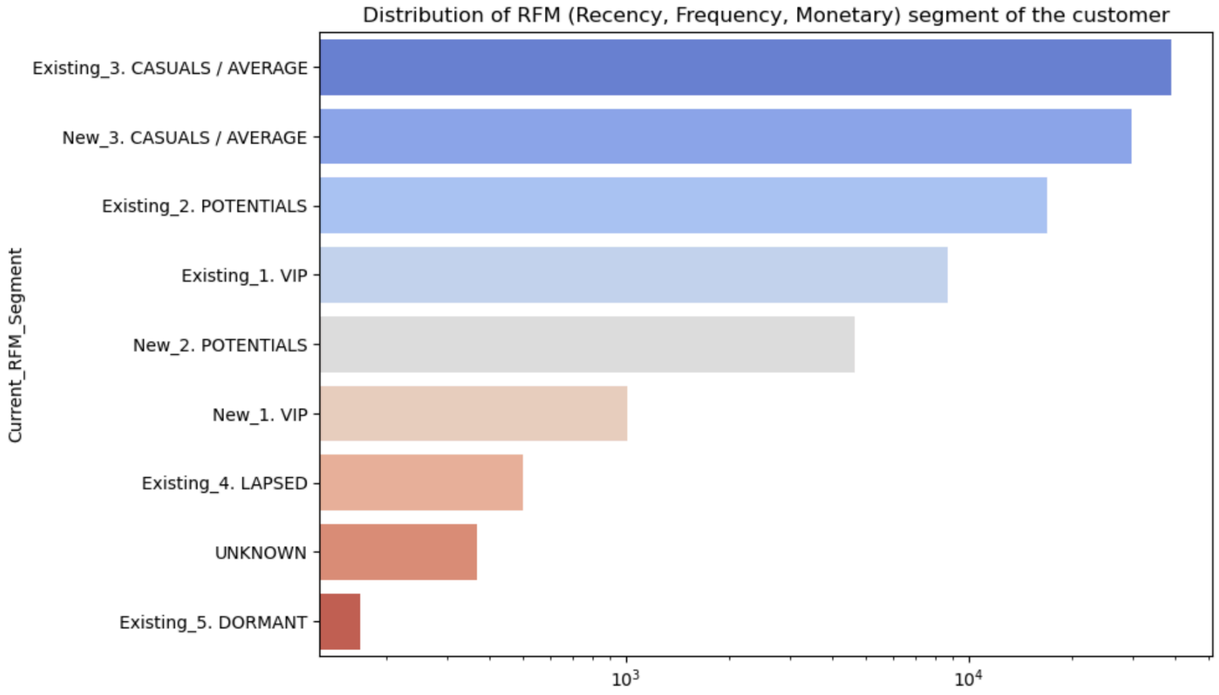
- To see the impact of variables on target variable (repurchase rate), checked several scatter plots.

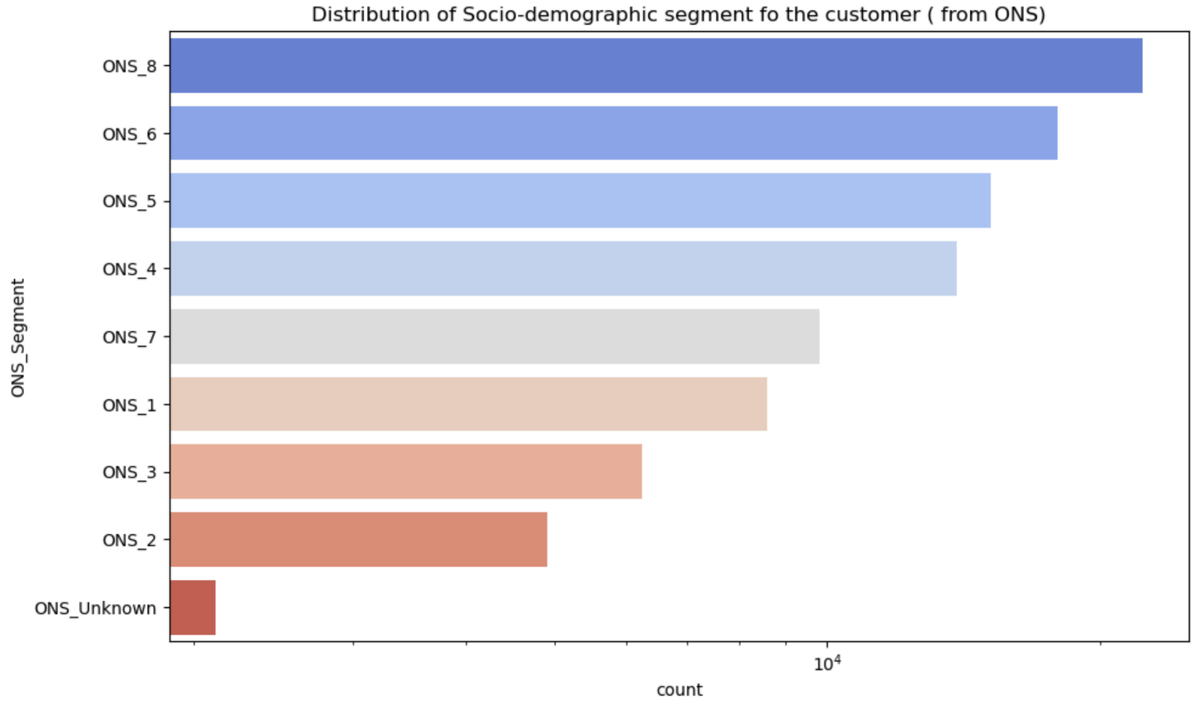



**For example, here proportion of Reduced Items doesn't show any relation on repurchase rate.**

# Exploratory analysis - III

- Insights from Categorical variables



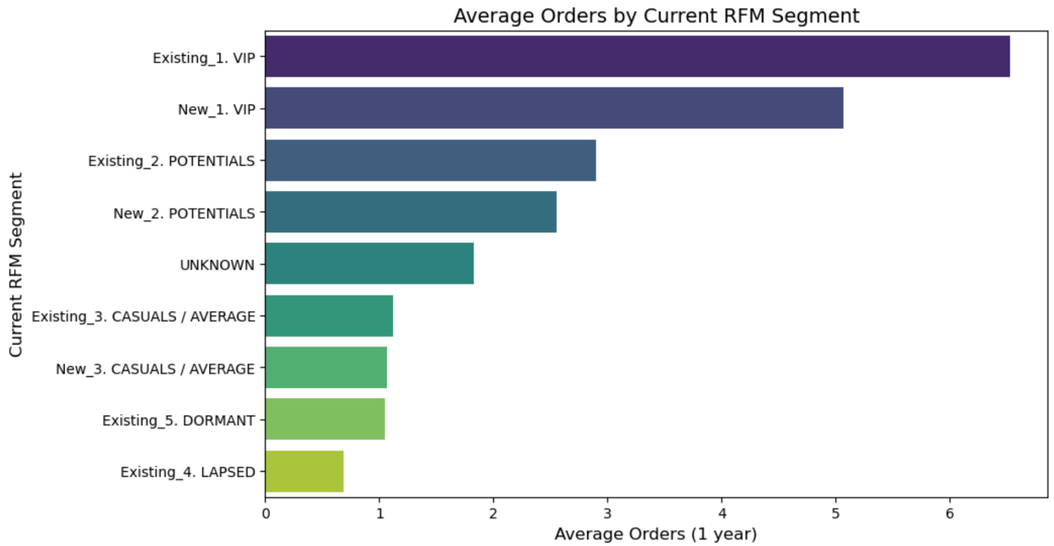

- **Current RFM Segment:**
  Certain RFM segments, such as "Casuals"/"POTENTIALS"/ "VIPs" have higher frequency compared to others while there are customers belonging to Lapsed/Unknown/Dormant category but their statistics is limited.
- **ONS Segment :**
  Certain ONS segments with higher engagement (e.g., "ONS_8"/"ONS_6") have higher frequency compared to others while here also we have customers belonging to Unknown category.

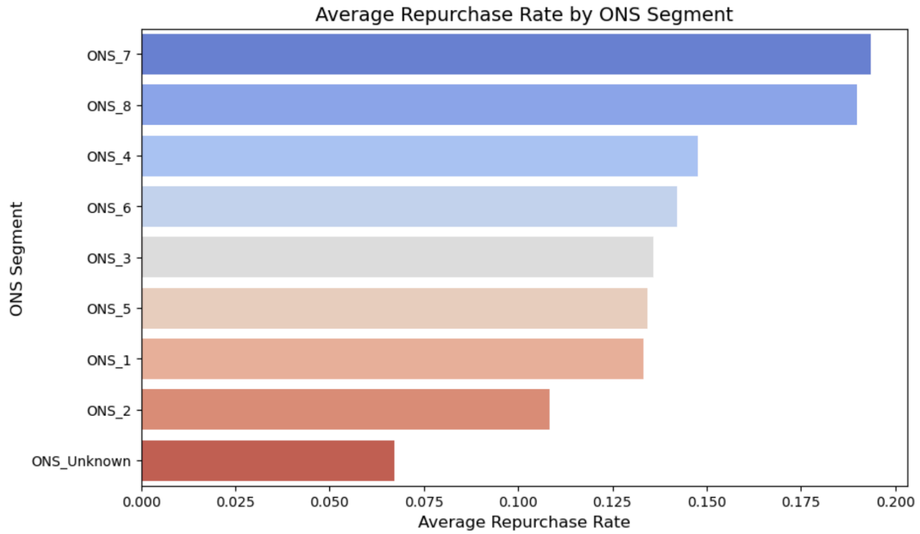
# Exploratory analysis - IV

Now I wanted to see:

- If there is any specific customer segmentation (a particular segmented customer based on categories have an impact on repurchase rate )
- If there is any promotional offer preference among customers
- Similary, if there is any brand/device preference between customers

# Insights from Customer Segmentation

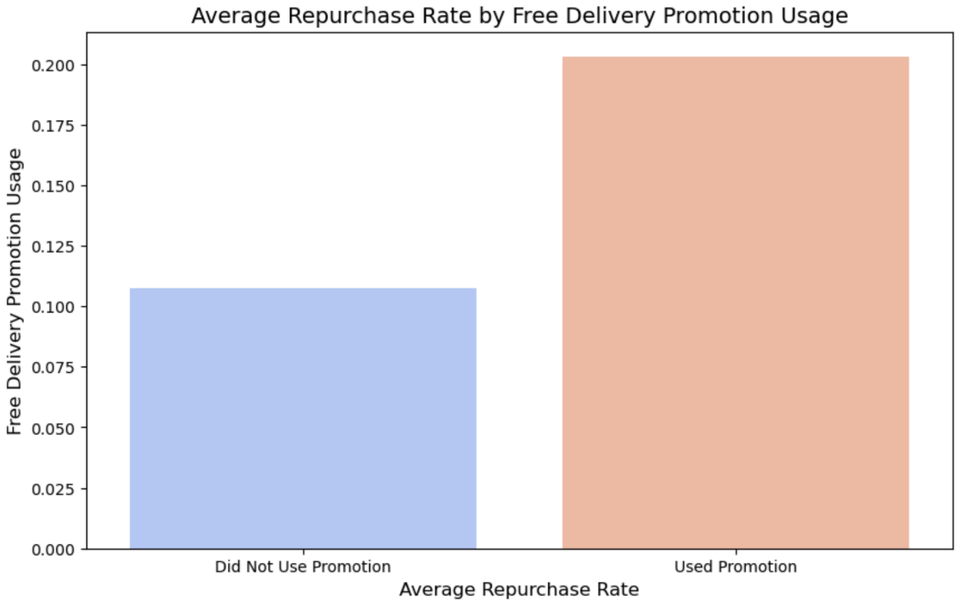


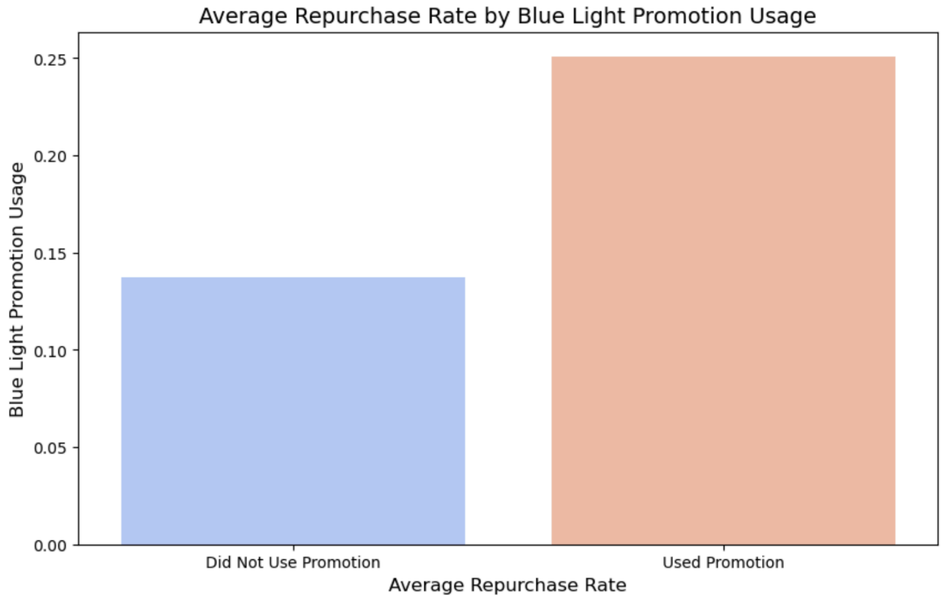




- **RFM Segment Analysis:**
  - Average Orders: Certain RFM segments, such as "VIPs"/"POTENTIALS," have higher average orders compared to others like "CASUALS." There is an unknown categoery which has higher average than causals.
  - Repurchase Rate: Segments like "VIPs" and "POTENTIALS" also show a higher repurchase rate, indicating that customers with higher engagement are more likely to repurchase.

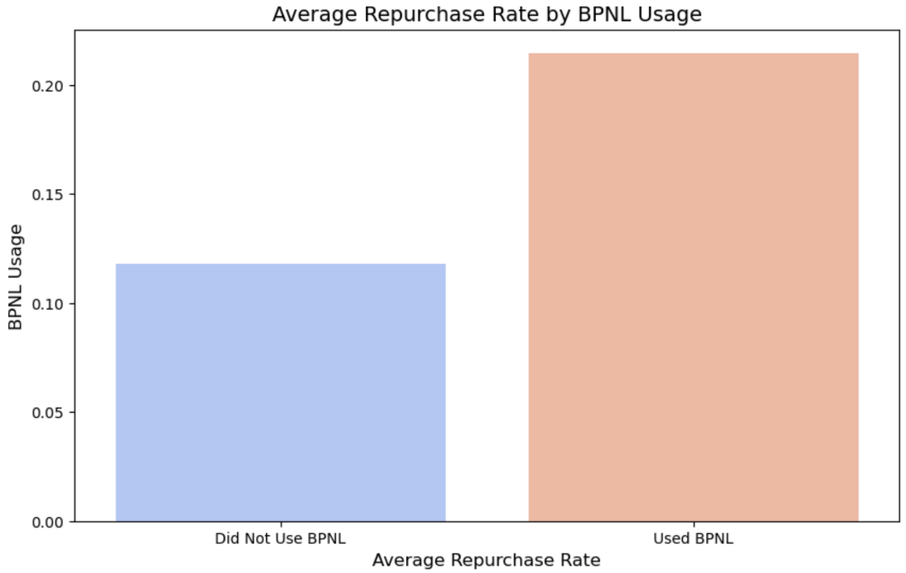
- **ONS Segment Analysis:**
  - Average Orders: ONS segments with higher engagement (e.g., "ONS_7" and "ONS_8") tend to have more orders compared to lower-engagement segments.
  - Repurchase Rate: Similarly, segments with higher repurchase rates align with more frequent orders.

# Insights from Promotional/BPNL Influence

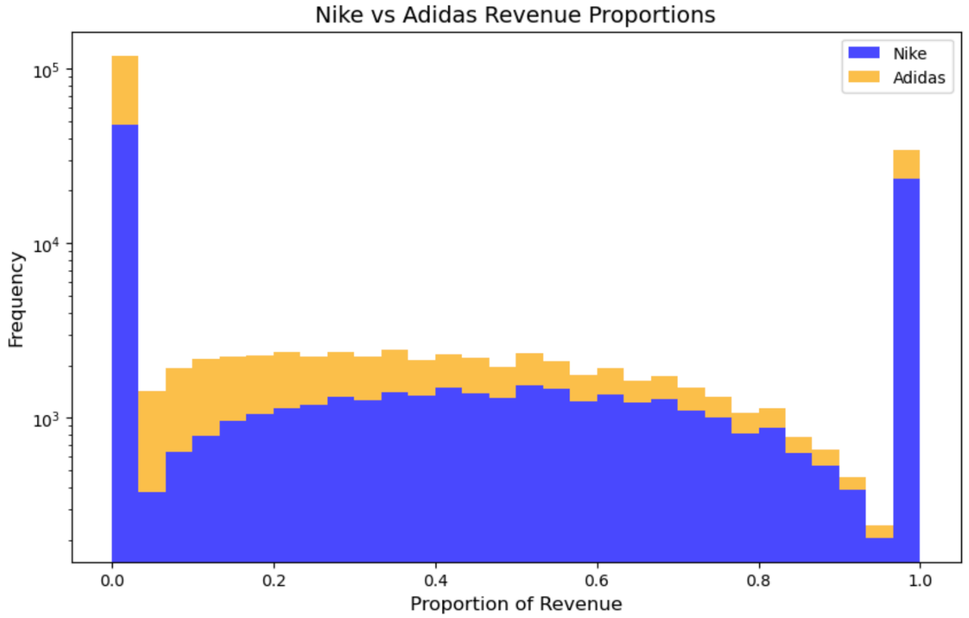


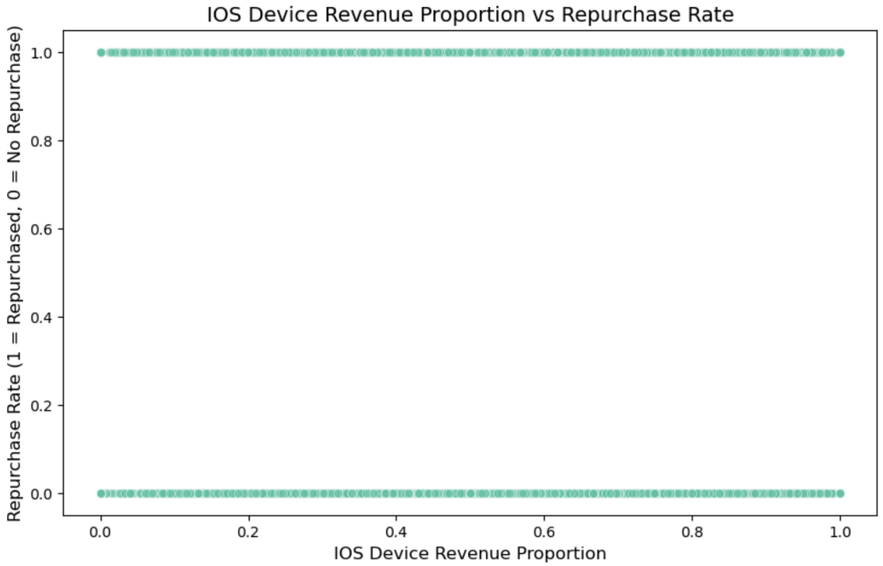



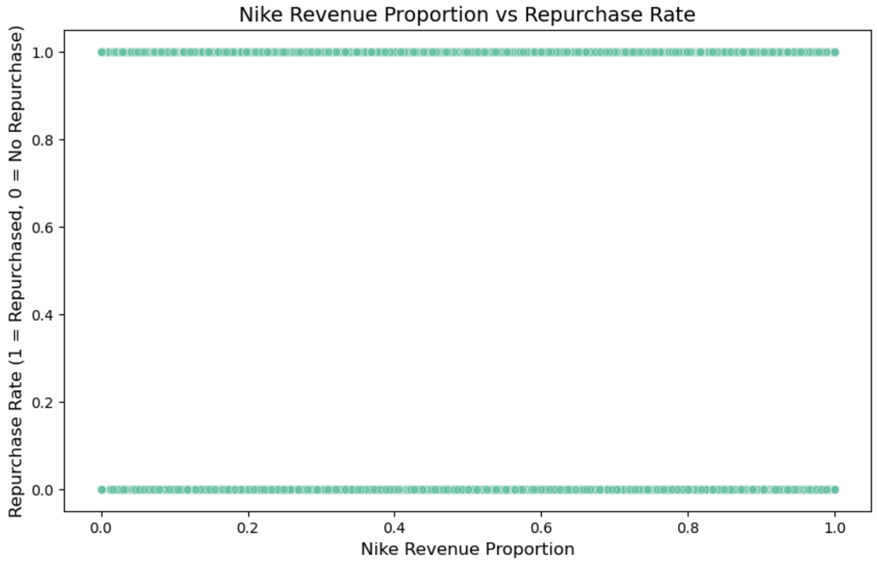

- **Repurchase Rate and Free Delivery Promotion:**
  The repurchase rate is slightly higher for customers who used the free delivery promotion compared to those who did not, suggesting a modest positive influence on repeat purchases.
- **Repurchase Rate and Blue Light Promotion:**
  Similarly, the repurchase rate is slightly higher for customers who used the Blue light promotion compared to those who did not, suggesting a modest positive influence on repeat purchases.
- **Repurchase Rate and BPNL:**
  Similarly, the repurchase rate is higher for customers who used the Buy Now Pay Later scheme compared to those who did not, suggesting again a positive influence on repeat purchases.
- **Dependence of these of promotions on number of orders was also seen and noticed similar behaviour.**



# Insights from Brand/Device Influence



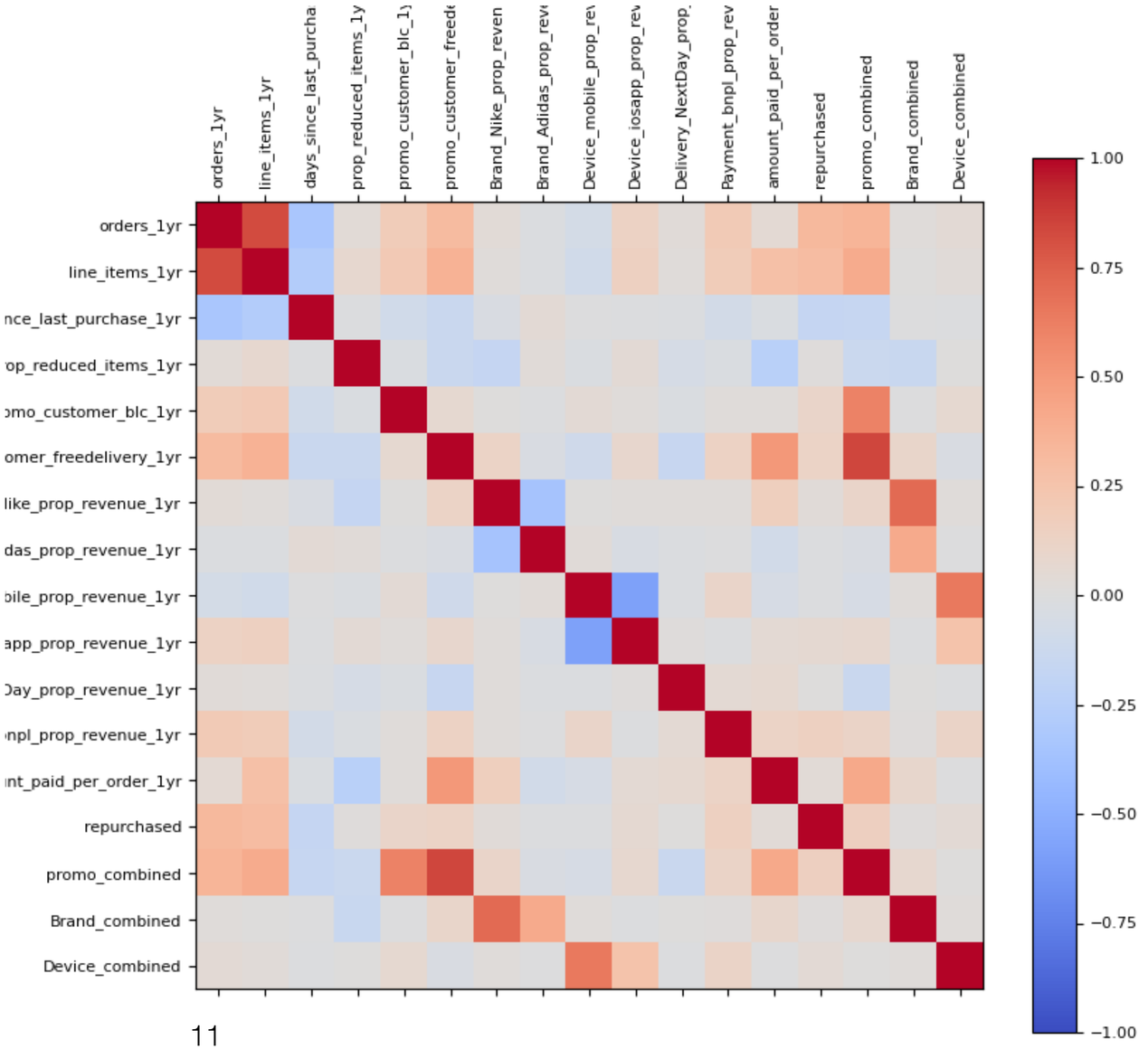




- **Nike vs. Adidas Revenue Proportions:**
  There are distinct customer segments with a clear preference for either Nike or Adidas, with some overlap. Customers appear more polarized towards one brand rather than splitting purchases evenly.
- **Nike Revenue vs. Repurchase Rate:**
  A higher proportion of revenue from Nike doesn't seem to strongly correlate with repurchase behavior Similar to Nike, no clear trend is seen for Adidas.
- **Device Revenue vs. Repurchase Rate:**
  Similar to Nike/Adidas, there is no clear trend indicating that a higher ios/mobile/nextday delievery revenue proportion drives repurchase behavior.

# Exploratory analysis - V

## Correlation Analysis:

- **Orders and Line Items:** Strong positive correlation indicating that customers who place more orders tend to purchase more items.
- **Days since last purchase Vs Orders/Line Items**
  There is a negative correlation between Days since last purchase and number of orders placed or line item, indicating inverse relation as expected.
- **Brands and Devices:**
  The correlation between brand usage (nike/adidas) is negative as one would expect, suggesting people who have a favourite brand say nike/they don't tend to spend on adidas and so forth. Simlarly, it's true between Device revenue proportions of mobiles and ios users.
- **Promotions and Orders:**
  The correlation between promotional usage (promo_customer_freedelivery_1yr, promo_customer_blc_1yr) and orders is positive, suggesting that promotions encourage higher purchasing activity.
- **Repurchase Rate:**
  **Weak** correlation with most variables, indicating that factors like promotional use or the amount paid per order may not strongly predict repeat purchases.



# Identify the Modelling

- Since we need to classify the customers in yes or no, its a binary classification exercise.
- Tried making 2 models one with Logestic Regression, and another with deep neural networks

# Approach

- Preprocess the data
    - Identify input features for the model
    - Remove the outliers
    - Scale the data
    - Deal with class imbalance
- Make the model
- Results

# Remember the hypothesis

- 0: when customer doesn’t repurchase, 1 : when customer repurchases
    - **TP :** "when customer doesn't make a purchase and has been predicted so (0,0)"
    - **FN** : "when customer actually made no purchase but model predicted he did (0,1)"
    - **FP** : "when customer actually made a purchase but model predicted he didn't (1,0)"
    - **TN** :  "when customer actually made a purchase but model predicted he did (1,1)”

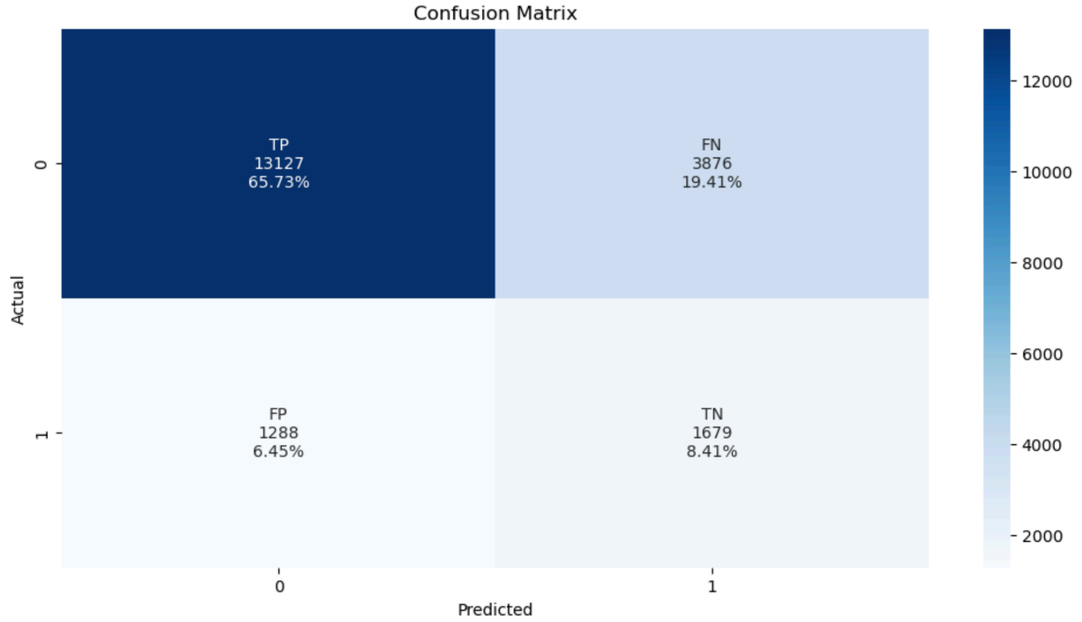
**Therefore, customers that the model will send emails to is where model predicts 1, which will be total of FN and TN. And to know, how many times my model predicted that customer would repurchase but in reality he wont (This is FN).**

**Idea is to maximise TP and TN while minimising FN and FP**

# Logistic Regression Model

# Results

```
Total Accuracy: 0.74
In a test sample of size 19970
This Model will send total emails : 5555
Where Estimated misclassification cost (in Pounds) is : 7752.00
                                Coefficient
orders_1yr                         0.662523
promo_combined                     0.131058
Brand_combined                     0.020959
Device_combined                    0.104054
Payment_bnpl_prop_revenue_1yr      0.219925
```




**Input Features**

```
dfc['promo_combined']= dfc['promo_customer_freedelivery_1yr']+dfc['promo_customer_blc_1yr']
dfc['Brand_combined']= dfc['Brand_Adidas_prop_revenue_1yr']+dfc['Brand_Nike_prop_revenue_1yr']
dfc['Device_combined'] = dfc['Device_mobile_prop_revenue_1yr']+dfc['Device_iosapp_prop_revenue_1yr']
```

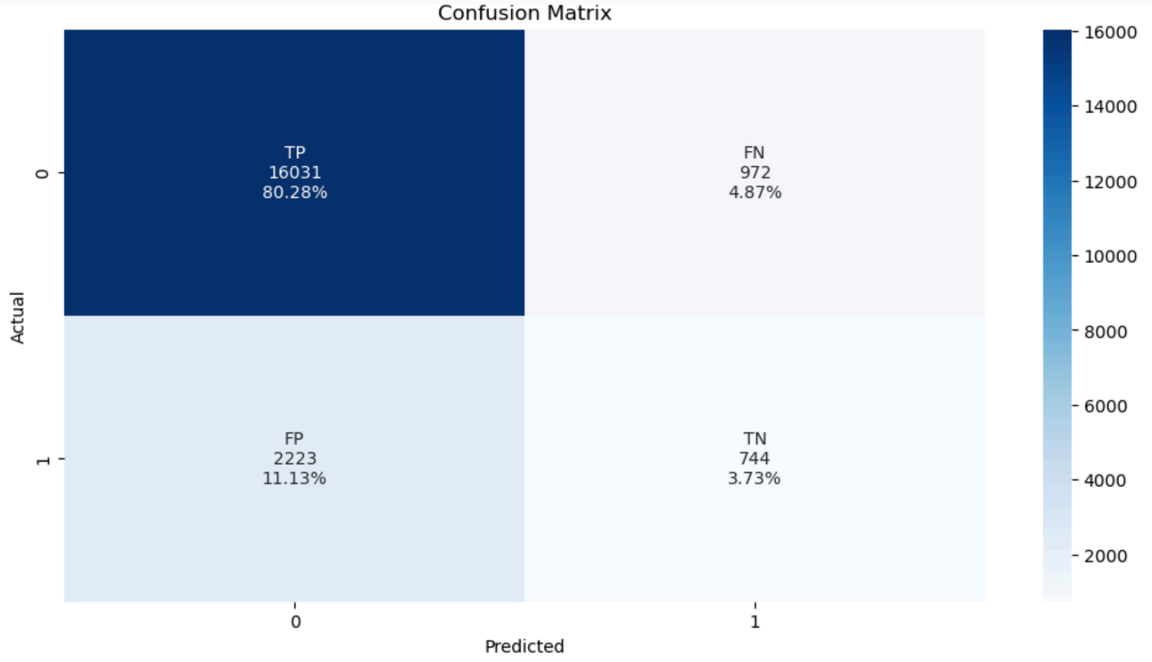
- **Started with only int/floats as inputs to the model!**
- **Decided to do simple submission of these variables as they are all negatively correlatively among themselves - to avoid multicollinearity issue**
- **Removed other variables based on various iterations of training and realising their less significance in the fit.**

# Insights from Logistic Regression Modelling

- **Here, accuracy is 74%**. Started by 64% of accuracy initially, improved it to 74% by :
  - Making sure there is no class imbalance
  - There are no outliers beyond 5 sigma (for simplicity, ofcourse one just can't decide the remove them blindly!), but wanted to make sure to have a working model first.
  - No categorical inputs are provided (One Hot encoding is kept for later)
  - Making sure the input featured are scaled using standard scaler to ensure that features are on the same scale.
  - Did basic feature engineering:
    - By hit and trial while training, noticed which variables are important, removed variables which had very low coefficients and hence were not important.
    - To deal with correlated variables provided their sum as input - again to keep things simple.
- **MissClassification cost comes out to be 7752.00 pounds**
- **Confusion matrix has a lot of mis-classification**, one could target to improve this further using more sophisticated feature engineering/adding regularisation/looking for over-fitting.
- **Therefore, tried GridSearchCV algorithm next**.

# Logistic Regression Modelling with GridSearchCV Algorithm

```
Best Hyperparameters: {'logreg__C': 0.1, 'logreg__class_weight': 'balanced', 'logreg__penalty': 'l2', 'logreg__solver': 'liblinear'}
```




```
Total Accuracy: 0.84
In a test sample of size 19970
This Model will send total emails : 1716
Where Estimated misclassification cost (in Pounds) is : 1944.00
```

```
/opt/anaconda3/lib/python3.11/site-packages/sklearn/model_selection/_validation.py:378: FitFailedWarning:
25 fits failed out of a total of 100.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.

Below are more details about the failures:
```

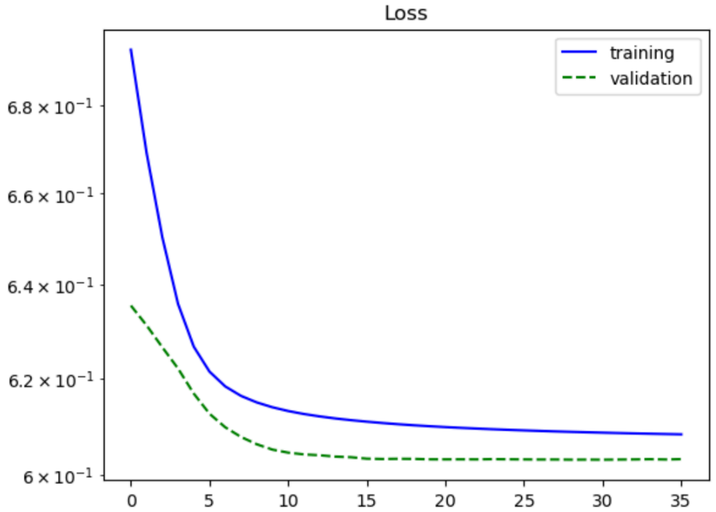
**Key Insights from Logestic Regresssion modelling**

- **Accuracy is 84%** with grid search algorithm **but fit has a warning of failing.**
- Confusion matrix is also better.
- Added regularization in grid-search but don't have much experience on how to check/ control overfitting in Logestic regression Modelling. Even though better results but need understanding of fit fail and thus confident in providing these numbers.
- Therefore at this point, decided of shifting to DNN binary classification where I have more experience from past to tune the model with hyper-parameter tuning and more control over how to look for any bias with overfitting.

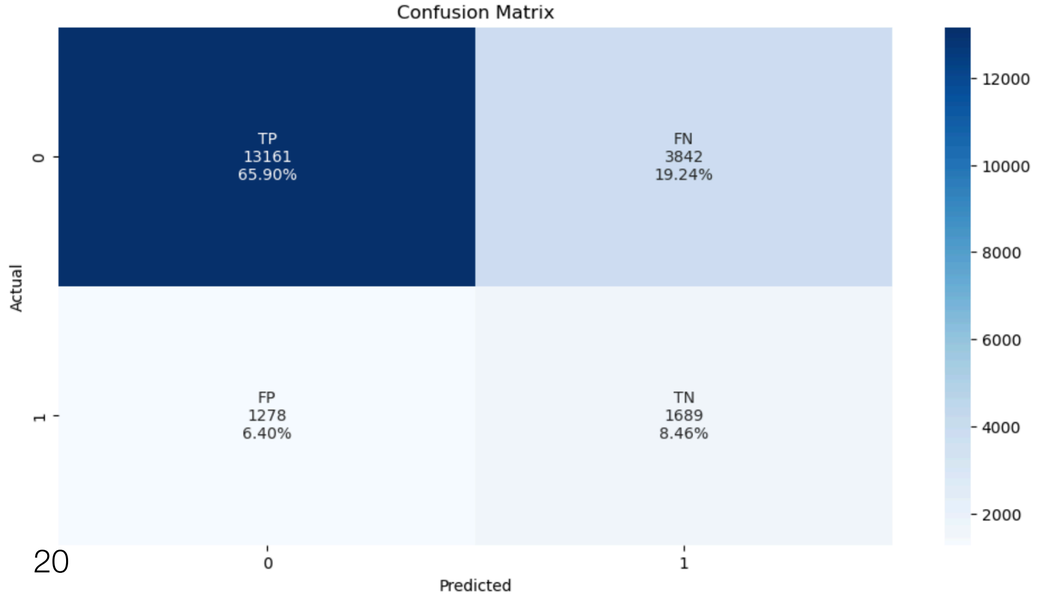
# DNN Binary Classification Model

# Model Architecture and Results

| Model Architecture | |
|---|---|
| Neurons | 64 |
| Layers | 3 |
| Optimizer | Adam |
| Loss Function | BinaryCrossentropy |
| Activation Function | Relu/Sigmoid |
| Learning Rate | 0.00001 |




```
Total Accuracy: 0.74
In a test sample of size 19970
This Model will send total emails : 5531
Where Estimated misclassification cost (in Pounds) is : 7684.00
```

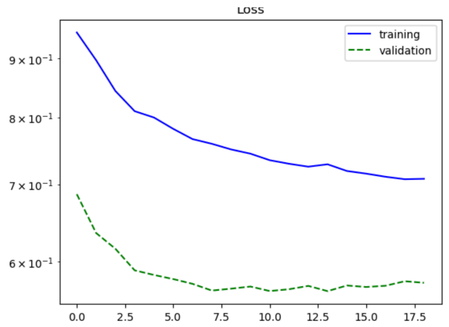


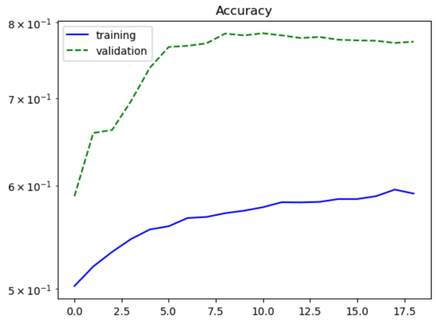
# Key Insights

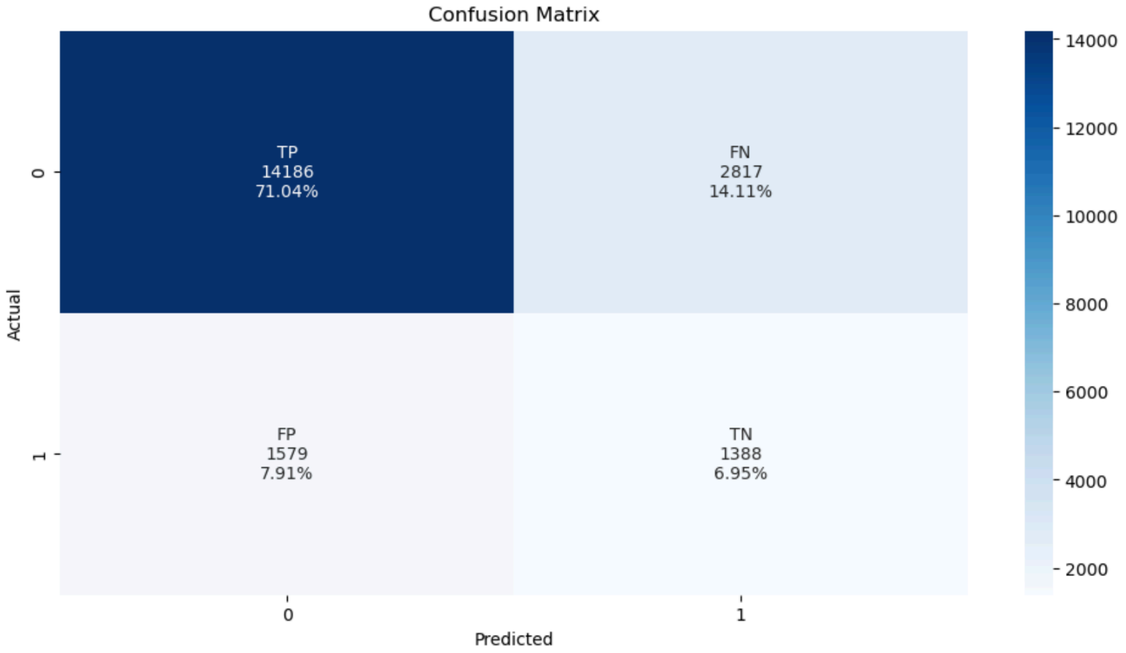
- Started with input features that gave good results with logestic regression model.
- For pre-processing used same caveats of removing outliers/scaling.
- Here for class imbalance, the technique of using class-weights as I did in regression model or using undersampler didnt work, so undersampled the imbalance using python only.
- Took several iterations to find "goodish" model using keras where there is no bias when we look at how loss function behaves as a function of epochs on test data. Accuracy is not smooth but probably adding more ways to avoid overfitting can help.
- Here, to deal with regularization, added early stopping in the model.
- **But overall, here the accuracy is 74% (74%) where in misidentification cost is slightly better 7684(7752) pounds (compared to logestic regession model without gridSerachCV algo where fit failed.)**
- Next is to see the effects of adding Batch normarmalization/DropOuts, if they make any impact here!

# Updated Model with Batch Normalisation and Dropouts

| Model Architecture | |
|---|---|
| Neurons | 64 For Input Layer (32 for Hidden layers) |
| Layers | 3 |
| Optimizer | Adam |
| Loss Function | BinaryCrossentropy |
| Activation Function | Relu/Sigmoid |
| Learning Rate | 0.00001 |



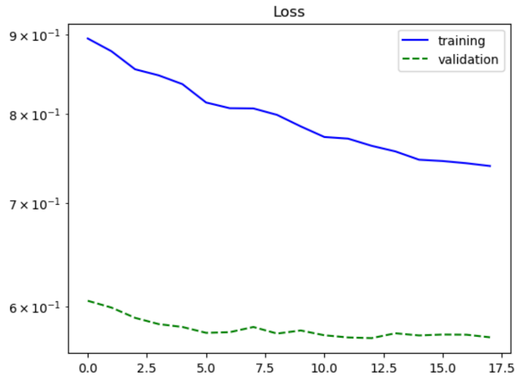

```
Total Accuracy: 0.78
In a test sample of size 19970
This Model will send total emails : 4205
Where Estimated misclassification cost (in Pounds) is : 5634.00
```

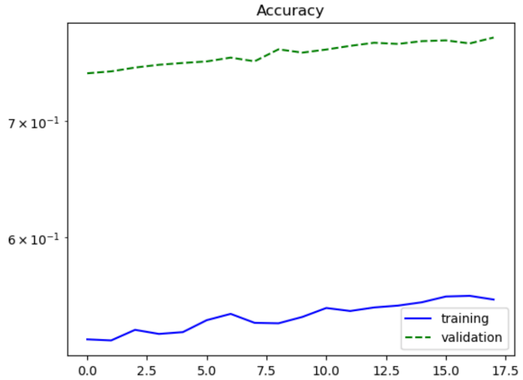
**Key Insights**

- Implemented the model again and look several iterations to find better model using sequential model.
- Here, to deal with overfitting and bias, added Batch Normalization and dropout after each layer in the model.
- **But overall, here the accuracy is slightly better 78% (74%) where in misidentification cost is better 5634 (7684) pounds (compared to without BN and DO techniques incorporated)**
- Next, is to see the impact of categorical variables during the training.

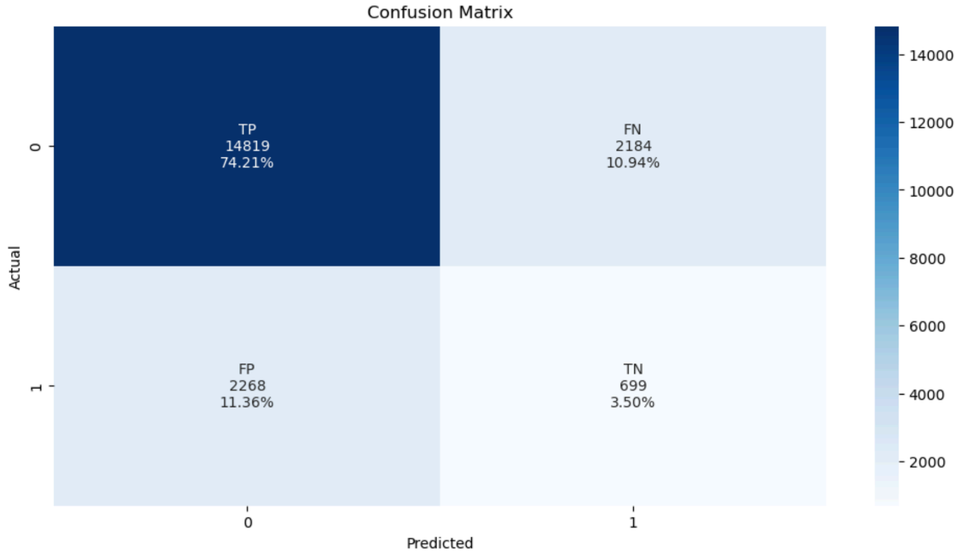
# Added Categorical Variables using One hot encoding as input features too!

| Model Architecture | |
|---|---|
| Neurons | 16 For Input Layer (16 for Hidden layers) |
| Layers | 3 |
| Optimizer | Adam |
| Loss Function | BinaryCrossentropy |
| Activation Function | Relu/Sigmoid |
| Learning Rate | 0.00001 |




**Here, the accuracy is 78% (78%) where in misidentification cost reduced to 4368 (5634) pounds (compared to without categorical variables)**




```
Total Accuracy: 0.78
In a test sample of size 19970
This Model will send total emails : 2883
Where Estimated misclassification cost (in Pounds) is : 4368.00
```

# Summary and Next Steps

**Best performance of DNN model predicting 78% accuracy and misclassification cost of ~4400 pounds in ~20000 test customers data**

**In Simple Logestic regression**

- **One could investigate the fit failure in logestic regression model with GridSearchCV algorithm**
- **Understand how to investigate overfitting**
- **Investigate more sophisticated feature engineering with correlated variables.**
- **Investigate impact of  categorical variables here as well**
- **Investigate outliers, currently just taken out for simplicity**

**DNN binary classification model**

- **More hyper-parameter tuning needed clearly looking at how loss function/accuracy behaves as function of epochs!**
- **More sophisticated feature engineering with correlated variables as for Simple Regression model**
- **More work on class imbalance techniques as currently the training data got under-sampleed based on minority class in output class.**
- **Again Investigate outliers!!**

# JupyterNotebooks

- Used python and jupter-notebook to look at the data
- JDExercise.ipynb - Used for exploratory and Logistic Regression analysis
- JDModel.ipynb - Used for DNN Binary Classification without Categorical variables (saving 2 models here!)
- JDModel_WOHE.ipynb - Used for DNN Binary Classification with Categorical variables used as input features to DNN (saving the last model as trained_model3.keras here, which can be used on fresh data)