IME672: Course Project

User Review Classification Problem

Group No 19

Tanishq Patil (170750)
Richeek Awasthi (160566)
Sajal Chaurasiya (170609)
Madhur Deep Jain (170367)

INTRODUCTION

Given the device used, browser used and the user review text, we are tasked with classifying if the review is positive or negative. We are provided with the following datasets-

1. **train.csv** - Training dataset with 38932 entries, each consisting of 'User_ID', 'Description', 'Browser_Used', 'Device_Used' and 'Is_Response'.
2. **train_clean1.csv** - Pre-Cleaned version of the training dataset with key words from user reviews replacing word descriptions.
3. **test.csv** - Test dataset with 29404 entries, each consisting of 'User_ID', 'Description', 'Browser_Used', 'Device_Used'. Since labels are missing, we can only benchmark the time taken to run over the test set.
4. **test_clean1.csv** - Pre-Cleaned version of the test dataset.

Our approach can be broken down into the following steps-

1. Data Exploration
2. Data Pre-Processing
3. Training Data on Selected Algorithms
4. Deriving Insights and Choosing Best Model

Data Exploration

The dataset has three pre-given attributes, names browser used, device used and description. In order to get some insight into the data, we'll look at each of these one by one.

Browser Used

Each user review belongs to one of **six distinct browsers** namely:

1. Internet Explorer
2. Google Chrome
3. Mozilla Firefox
4. Opera
5. Safari
6. Edge

There was some duplication in some browser names like

1. Internet Explorer - InternetExplorer - IE
2. Google Chrome - Chrome
3. Mozilla Firefox - Mozilla - Firefox

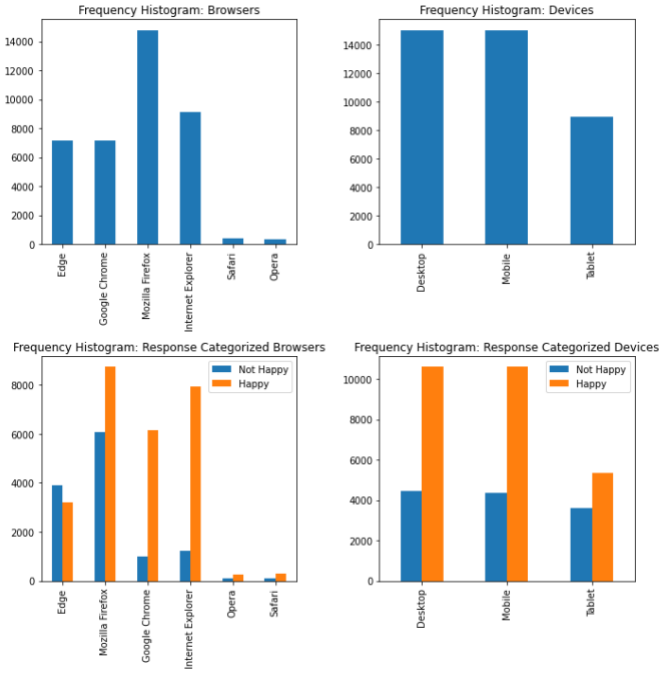
This was fixed and visual exploration of the number of positive and negative reviews stemming from each of these browsers was done. The results are affixed below.

Device Used

There are three categories in this field namely:

1. Desktop
2. Mobile
3. Tablet

Frequency histograms corresponding to each of these devices and the split across happy and unhappy reviews are given below.

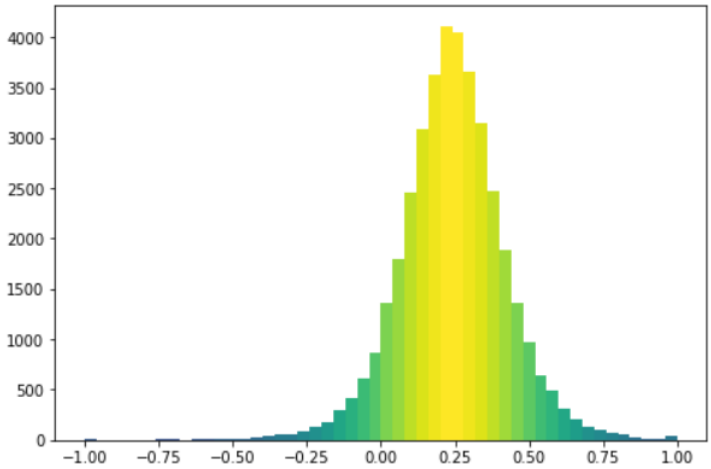


Description

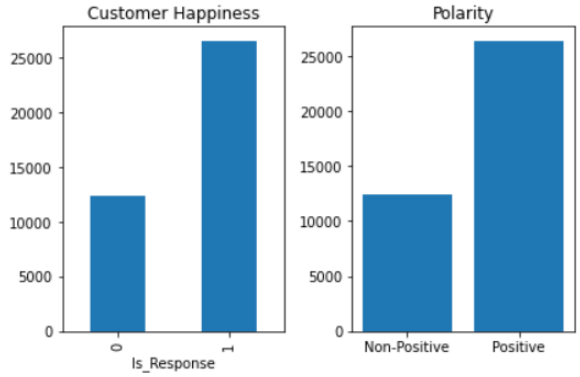
Going through the review texts, the following insights were found-

Number of words: 2595870
Number of unique words: 52409
Number of Happy Words used: 970469
Number of Non_Happy Words used: 1625401
Number of Unique Happy Words used: 28196
Number of Unique Non_Happy Words used: 38444
Average No. of words in Happy Descriptions: 78.19426315365402
Average No. of words in Non_Happy Descriptions: 61.28731948267411

The facts found above lead us to believe that analysing the sentiment of words used in these descriptions is key to classifying the reviews. For the aforementioned task, we use **TextBlob**, an open source library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. Using this on distinct words in descriptions, we obtain the following frequency distribution against sentiment scores, 1 being most positive and -1 being most negative.



Considering that the polarity of words is around 0.17 on an average, we tried splitting training samples on this criterion and compared this split to the split by labels, and we witnessed strong correlation between the two as seen below.



Later, another polarity measure, namely VADER was also used, which is described below.

Data Pre-Processing

In order to train the model reliably, we need to process the attributes in a trainable format and construct training & test variables. In order to do so we define the variables we'll consider for classification-

1. **Browser Used** - With each browser getting an attribute, the browser used corresponding to a particular entry will have 1 as its value.
2. **Device Used** - Same as Browser Used.
3. **TextBlob Polarity** - TextBlob provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more
4. **Vader Polarity** - VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.

5. **Word Count (removed later)** - Number of words in a description, considered due to a difference in the average no of words used in negative and positive reviews, removed to witness an improvement in model's performance.

Since the test set provided at Kaggle does not contain labels, we split the training data using a 60-20-20 split into training, validation and testing sets.

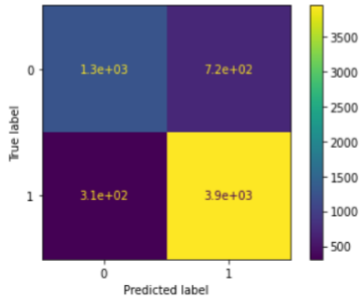
Model Training

We'll be training the data on a set of algorithms ranging from simple models like Logistic Regression, Support Vector Machine etc. to more complex models like AdaBoost Classifier, XGBoost Classifier etc.

Logistic Regression

	precision	recall	f1-score	support
0	0.80	0.64	0.71	1974
1	0.85	0.93	0.88	4255
accuracy			0.83	6229
macro avg	0.82	0.78	0.80	6229
weighted avg	0.83	0.83	0.83	6229

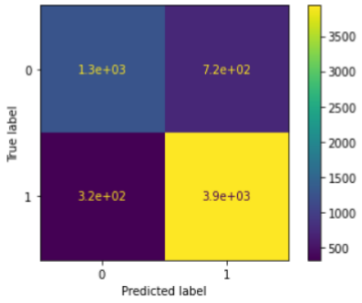
```
Time Taken: 0.19652652740478516
ROC Score: 0.7811022136184025
```



Logistic Regression (With Cross-Validation)

	precision	recall	f1-score	support
0	0.80	0.64	0.71	1974
1	0.85	0.93	0.88	4255
accuracy			0.83	6229
macro avg	0.82	0.78	0.80	6229
weighted avg	0.83	0.83	0.83	6229

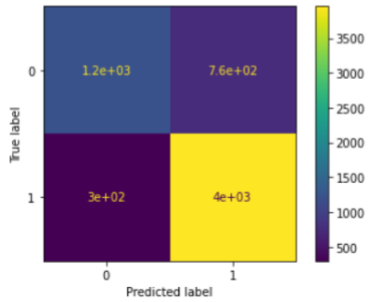
```
Time Taken: 2.583468437194824
ROC Score: 0.7814103319653736
```



Support Vector Classifier

	precision	recall	f1-score	support
0	0.80	0.62	0.70	1974
1	0.84	0.93	0.88	4255
accuracy			0.83	6229
macro avg	0.82	0.77	0.79	6229
weighted avg	0.83	0.83	0.82	6229

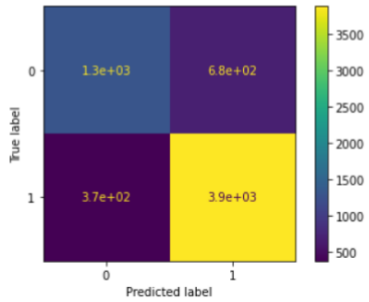
```
Time Taken: 19.360989332199097
ROC Score: 0.7728506423696063
```



AdaBoost Classifier

	precision	recall	f1-score	support
0	0.78	0.66	0.71	1974
1	0.85	0.91	0.88	4255
accuracy			0.83	6229
macro avg	0.81	0.79	0.80	6229
weighted avg	0.83	0.83	0.83	6229

```
Time Taken: 0.9237029552459717
ROC Score: 0.7852958019470508
```

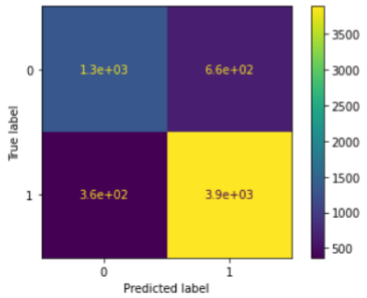


XGBoost Classifier

Multilayer Perceptron Classifier

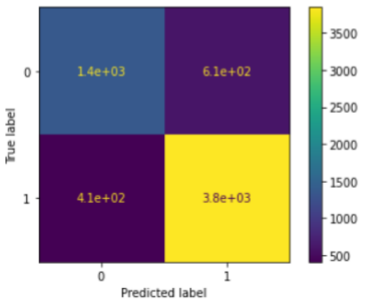
	precision	recall	f1-score	support
0	0.78	0.67	0.72	1974
1	0.86	0.91	0.88	4255
accuracy			0.84	6229
macro avg	0.82	0.79	0.80	6229
weighted avg	0.83	0.84	0.83	6229

Time Taken: 1.4653089046478271
ROC Score: 0.7910667109557027



	precision	recall	f1-score	support
0	0.77	0.69	0.73	1974
1	0.86	0.90	0.88	4255
accuracy			0.84	6229
macro avg	0.82	0.80	0.81	6229
weighted avg	0.83	0.84	0.83	6229

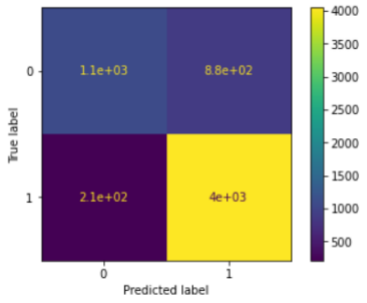
Time Taken: 10.729132413864136
ROC Score: 0.7979368690747044



Linear Discriminant Analysis

	precision	recall	f1-score	support
0	0.84	0.55	0.67	1974
1	0.82	0.95	0.88	4255
accuracy			0.82	6229
macro avg	0.83	0.75	0.77	6229
weighted avg	0.83	0.82	0.81	6229

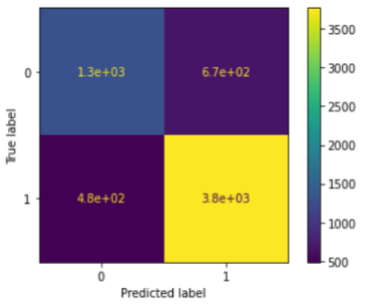
Time Taken: 0.07082271575927734
ROC Score: 0.7520364027302049



K Neighbours Classifier

	precision	recall	f1-score	support
0	0.73	0.66	0.69	1974
1	0.85	0.89	0.87	4255
accuracy			0.82	6229
macro avg	0.79	0.77	0.78	6229
weighted avg	0.81	0.82	0.81	6229

Time Taken: 0.521125078201294
ROC Score: 0.7743152164983803



Deep Neural Classifier

We used a **11->100->70->40->20->15->10->5->1** neural net architecture, using **RELU** activation for non-output layers and a **sigmoid** activation for the output layer. The loss function used is '**binary_crossentropy**' and the solver used is '**adam**'. The model was trained for 100 epochs with a batch size of 64. It converged with both the training and validation accuracy to be around 83%.

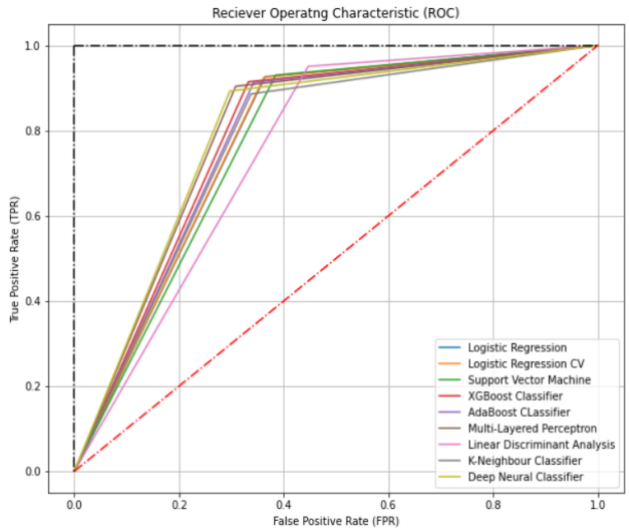
```
Time Taken: 80.86778593063354
              precision    recall  f1-score   support

         0.0       0.70      0.75      0.73      1845
         1.0       0.89      0.87      0.88      4384

    accuracy                           0.83      6229
   macro avg       0.80      0.81      0.80      6229
weighted avg       0.84      0.83      0.83      6229

[[1388  586]
 [ 457 3798]]
ROC Score: 0.7978688877856315
```

ROC Curves



Insights

1. Browsers like Edge, Internet Explorer, Mozilla Firefox, and Google Chrome had a significantly large number of customers. Among these, the users unhappy with Edge were more in number than the happy users.
2. For the devices used to access the browsers, the ratio of happy to unhappy customers was high for Desktops and Phones. For tablets, it was slightly greater than one.
3. Most customers, generally, tend to provide a description which was towards the positive side on a scale of [-1, 1], where -1 indicating a highly negative description and +1 indicating a highly positive one. Mode of the distribution was approximately 0.25.
4. The data given for the training was able to train the models with a maximum accuracy of 83%. None of the classifiers was able to obtain a classification accuracy of more than that.
5. From the given unseen data, almost 2/3rd of the customers were classified as happy. And as observed in the training set, most of the happy customers used one of the browsers from Internet Explorer, Google Chrome, or Mozilla Firefox. Many of them were either desktop users or mobile phone users.

Choosing the Best Model

In order to deploy this classifier into use, we must look at the following critical factors-

1. Time Taken
2. Accuracy
3. ROC Score

Considering these factors in mind, XGBoost appears to be the best model we've trained. It has an accuracy of 84%, which is the highest of all models. It also has the 3rd highest ROC scores among all models, after Multi-Layered Perceptrons and Deep Neural Networks, both of which take much more time to train and have lower accuracy as compared to XGBoost.

Hence, XGBoost is the recommended algorithm to be used for classifying the test set.