BELLEVUE
UNIVERSITY

Amazon Musical Instruments Reviews

Understand the Customer Feedback

Source: kaggle.com Dataset

DSC550 – Data Mining
Final Project – Case Study Report
Spring 2021
Manish Kalkar

Introduction

The Kaggle Dataset contains the customer feedback on *amazon.com* for Musical Instruments. The opinions expressed as part of customer feedback are in free text fields. The classification of individual comments / reviews from the customer ultimately determines overall rating given by the customer. Organization needs to get a complete idea on feedback provided by customers, so that it will enable the organization to develop more loyal customers, thereby increase in business, brand value and profits.

Business Problem

Going through all the customer feedbacks can be very tedious job because of feedbacks captured online are in open text fields. The customer feedback in the form of these free text fields is very hard to analyze in order to give proper justice to the issues related to musical instruments that customers are coming across. The feedback somehow needs to classify so that the feedback category can be identified based on the overall rating given by the customer. The categorized feedback then can be utilized to get a complete idea of customer sentiments towards the musical instruments' product line.

Without analyzing the customer feedback create challenges for the organization to develop strategies to boost revenue, retain customers and maintain brand value.

Objective

- Perform Sentimental Analysis on this dataset, thereby helping the organization to understand their customer feedback better. That way, organization can concentrate on the issues that customers are facing.
- Build the model which has highest accuracy and / or any other performance metric in classifying the feedback as positive, Negative, and Neutral.
- Predict the overall rating of the Musical Instrument based on the outcome of the model evaluation and the results of the performance metric.

The Data / Dataset

The dataset contains 10261 rows and 9 columns (attributes) with 1429 unique customers providing feedback on as many as 900 Musical Instruments. Each row corresponds to an individual customer review submitted online on Amazon.com.

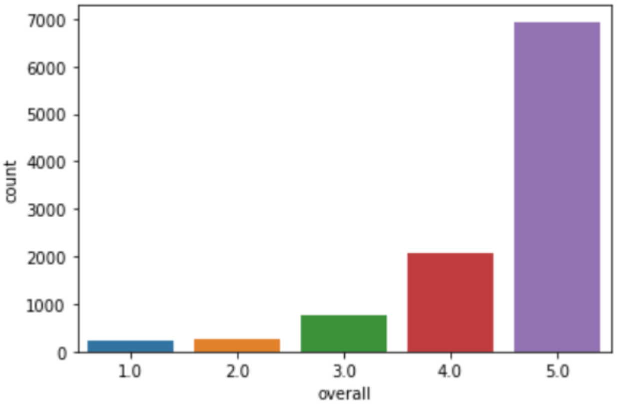
Data File: Musical_instruments_reviews.csv

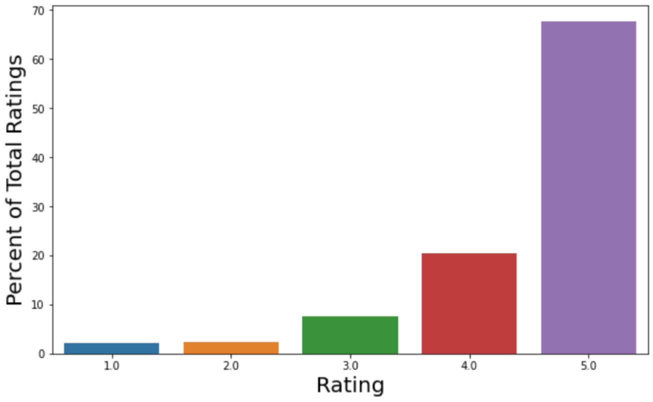
Dataset fields / attributes:

1. **reviewerID** - ID of the reviewer, e.g. A2SUAM1J3GNN3B
2. **asin** - ID of the product, e.g., 0000013714
3. **reviewerName** - name of the reviewer
4. **helpful** - helpfulness rating of the review, e.g., 2/3
5. **reviewText** - text of the review
6. **overall** - rating of the product
7. **summary** - summary of the review
8. **unixReviewTime** - time of the review (unix time)
9. **reviewTime** - time of the review (raw)

Graphical Analysis

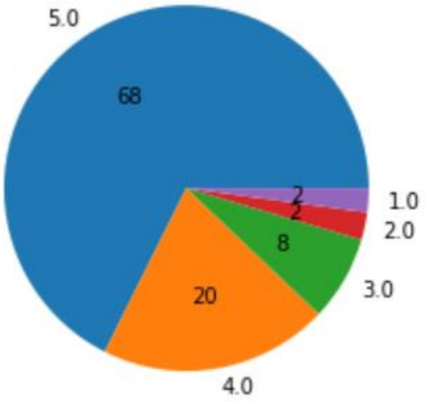
1. Plot the Total Number of Overall Ratings (Counts and percentages) of the customer reviews





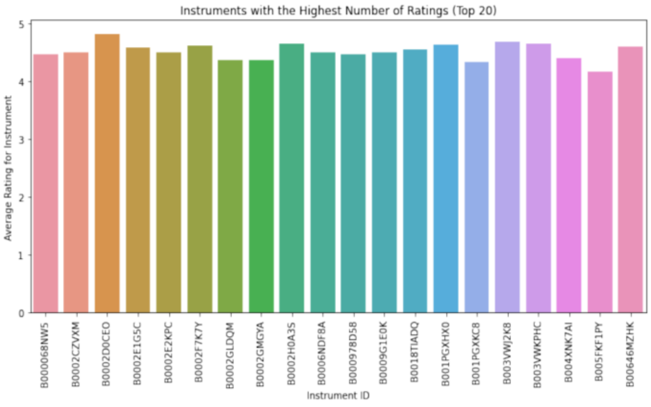
Overall Rating 5.0 has got highest counts as well as percentage followed by 4.0 and 3.0. The Overall Ratings of 2.0 and 1.0 belong to unhappy customers remain almost the same in terms of the percentages.

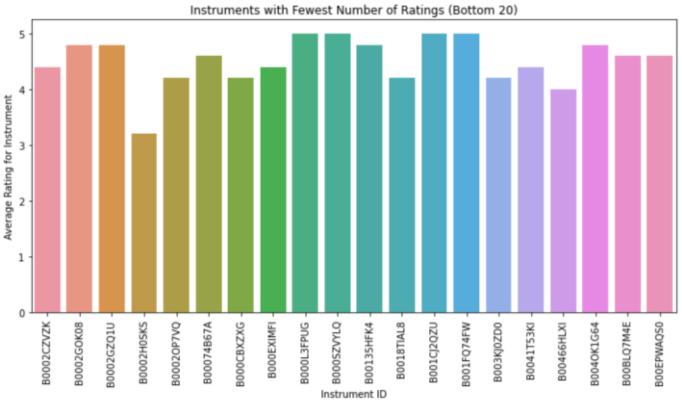
2. Plot the Pie Chart showing Overall Ratings – Percentages



Above Pie Chart shows the Overall Rating 5.0 has got highest percentage of 68%, the rating 4.0 has got 20% and 3.0 has got 8% of the Overall Rating. The Overall Ratings of 2.0 and 1.0 belong to unhappy customers remain almost the same - 2% each.

3. Plot the Musical Instruments with the Highest and Fewest Number of Ratings (Top 20 and Bottom 20)



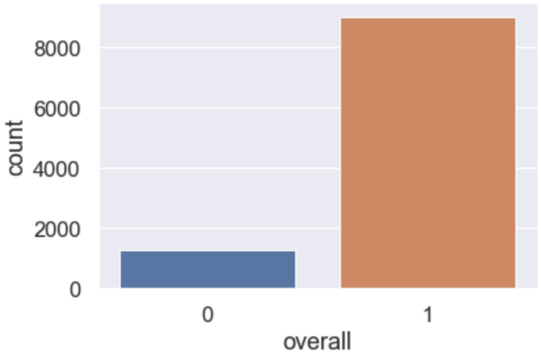


The dataset contains total 10261 reviews on amazon.com provided by 1429 customers on total 900 unique Musical Instruments. Left Plot shows Musical Instruments with the Highest Number of Ratings (Top 20) and Right Plot shows Musical Instruments with the Fewest Number of Ratings (Bottom 20). In both cases, Average Rating for the instruments have been plotted.

4. Plot simplified version of overall ratings of the customer reviews

Simplify Overall ratings by re-assigning the values as below:

- Overall Ratings 4 and above = 1
- Overall Ratings 3 and below = 0

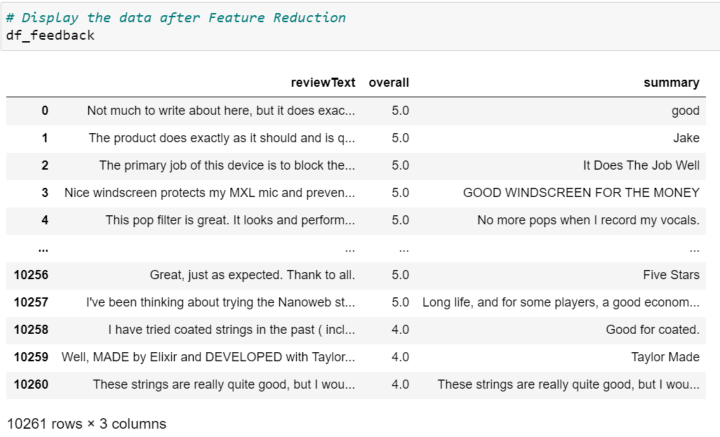


Simplified Rating	Count => Percentage
1	**9022 => 87%**
0	**1239 => 12%**

Dimensionality & Feature Reduction

List of features available and their categorization relevant vs. irrelevant

1. **reviewerID** – irrelevant feature – can be dropped
2. **asin** - irrelevant feature – can be dropped
3. **reviewerName** - irrelevant feature – can be dropped
4. **helpful** - irrelevant feature – can be dropped
5. **reviewText** – relevant feature – needed for the analysis
6. **overall** – relevant feature – needed for the analysis
7. **summary** – relevant feature – needed for the analysis
8. **unixReviewTime - irrelevant feature – can be dropped**
9. **reviewTime - irrelevant feature – can be dropped**



```
# Display the data after Feature Reduction
df_feedback
```

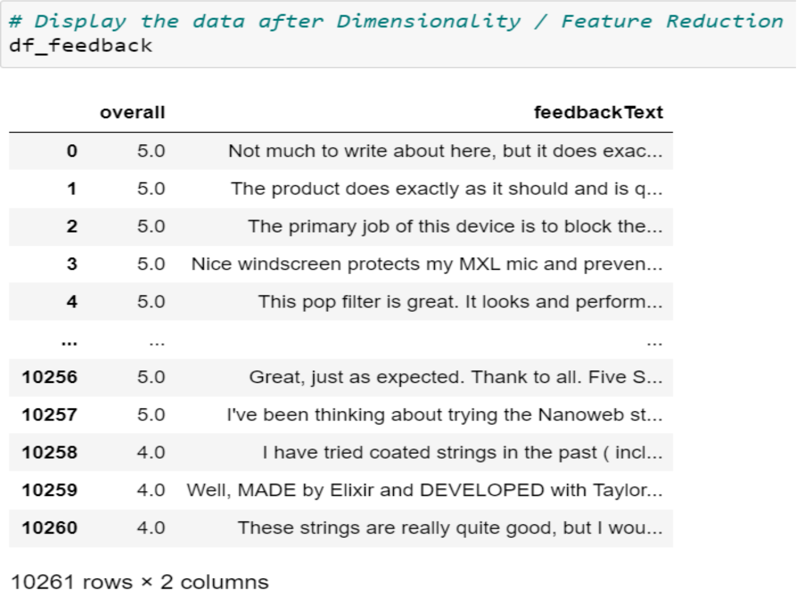
	reviewText	overall	summary
0	Not much to write about here, but it does exac...	5.0	good
1	The product does exactly as it should and is q...	5.0	Jake
2	The primary job of this device is to block the...	5.0	It Does The Job Well
3	Nice windscreen protects my MXL mic and preven...	5.0	GOOD WINDSCREEN FOR THE MONEY
4	This pop filter is great. It looks and perform...	5.0	No more pops when I record my vocals.
...	...	...	...
10256	Great, just as expected. Thank to all.	5.0	Five Stars
10257	I've been thinking about trying the Nanoweb st...	5.0	Long life, and for some players, a good econom...
10258	I have tried coated strings in the past (incl...	4.0	Good for coated.
10259	Well, MADE by Elixir and DEVELOPED with Taylor...	4.0	Taylor Made
10260	These strings are really quite good, but I wou...	4.0	These strings are really quite good, but I wou...

10261 rows × 3 columns

Feature Engineering

Below are the steps taken to further engineer the relevant features in order to make them ready for the analysis.

- Address any missing data issues
- Build any new features that are needed for the model
  - Build new feature **feedbackText** by combining two features **reviewText + summary**
  - Further reduce the features by dropping the features reviewText and summary as we do not need them anymore.



```
# Display the data after Dimensionality / Feature Reduction
df_feedback
```

	overall	feedbackText
0	5.0	Not much to write about here, but it does exac...
1	5.0	The product does exactly as it should and is q...
2	5.0	The primary job of this device is to block the...
3	5.0	Nice windscreen protects my MXL mic and preven...
4	5.0	This pop filter is great. It looks and perform...
...	...	...
10256	5.0	Great, just as expected. Thank to all. Five S...
10257	5.0	I've been thinking about trying the Nanoweb st...
10258	4.0	I have tried coated strings in the past (incl...
10259	4.0	Well, MADE by Elixir and DEVELOPED with Taylor...
10260	4.0	These strings are really quite good, but I wou...

10261 rows × 2 columns

Sentiment Analysis

Sentiment Analysis has been performed based on below listed methods:

1. VADER
2. textblob
3. Based on mapping of Overall Rating Score

Added below two new features / columns to accommodate sentiment data

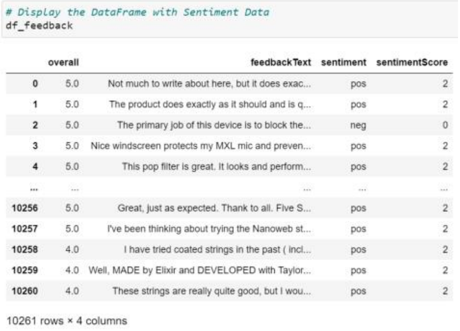
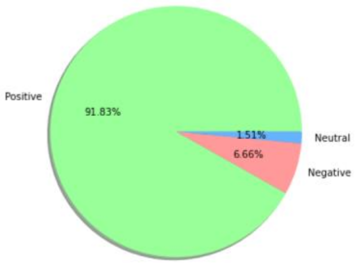
- sentiment
- sentimentScore

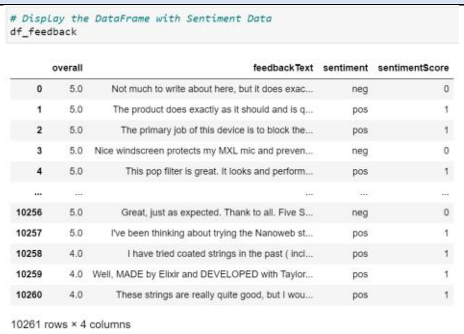
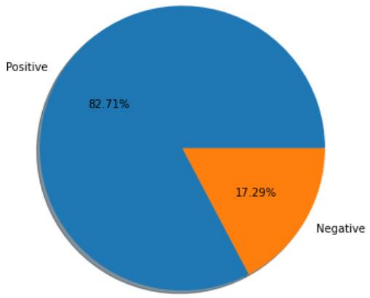
<table>
<tr><th>VADER</th><th>textblob Blobber and NaiveBayesAnalyzer</th><th>Based on mapping of Overall Rating Score</th></tr>
<tr><td><b>Measure Sentiment</b></td><td><b>Measure Sentiment</b></td><td><b>Measure Sentiment</b></td></tr>
<tr>
<td>-- sentiment - pos - Positive, neg - Negative, neu – Neutral

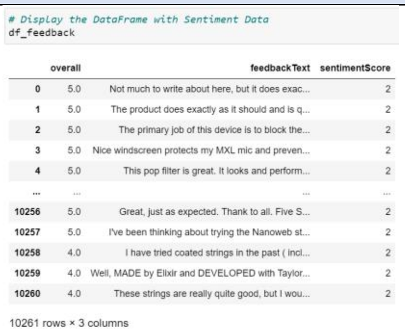
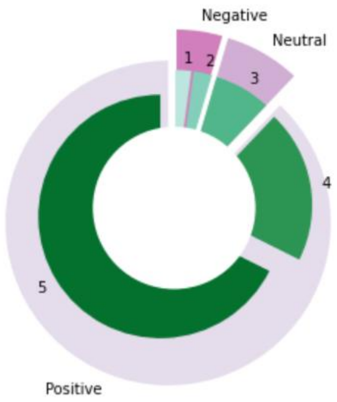
-- sentimentScore - 2 for Positive, 0 for Negative and 0 for Neutral</td>
<td>-- sentiment - pos - Positive, neg – Negative

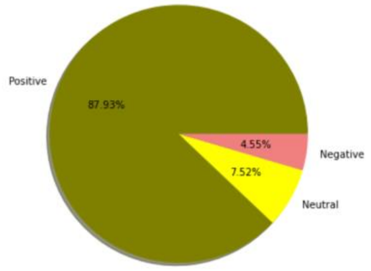
-- sentimentScore - 1 for Positive, 0 for Negative

-- Neutral and Negative categories are combined and considered as Negative Review category.</td>
<td><table><tr><td>O-R</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr><tr><td>S-S</td><td>2</td><td>2</td><td>1</td><td>0</td><td>0</td></tr></table>
O-R: Overall Rating
S-S: sentimentScore

Sentiment 2 - Positive
Sentiment 1 - Neutral
Sentiment 0 - Negative</td>
</tr>
<tr><td><b>Results</b></td><td><b>Results</b></td><td><b>Results</b></td></tr>
<tr>
<td>
Display the DataFrame with Sentiment Data
df_feedback
<table>
<tr><th></th><th>overall</th><th>feedbackText</th><th>sentiment</th><th>sentimentScore</th></tr>
<tr><td>0</td><td>5.0</td><td>Not much to write about here, but it does exac...</td><td>pos</td><td>2</td></tr>
<tr><td>1</td><td>5.0</td><td>The product does exactly as it should and is q...</td><td>pos</td><td>2</td></tr>
<tr><td>2</td><td>5.0</td><td>The primary job of this device is to block the...</td><td>neg</td><td>0</td></tr>
<tr><td>3</td><td>5.0</td><td>Nice windscreen protects my MXL mic and preven...</td><td>pos</td><td>2</td></tr>
<tr><td>4</td><td>5.0</td><td>This pop filter is great. It looks and perform...</td><td>pos</td><td>2</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>10256</td><td>5.0</td><td>Great, just as expected. Thank to all. Five S...</td><td>pos</td><td>2</td></tr>
<tr><td>10257</td><td>5.0</td><td>I've been thinking about trying the Nanoweb st...</td><td>pos</td><td>2</td></tr>
<tr><td>10258</td><td>4.0</td><td>I have tried coated strings in the past (incl...</td><td>pos</td><td>2</td></tr>
<tr><td>10259</td><td>4.0</td><td>Well, MADE by Elixir and DEVELOPED with Taylor...</td><td>pos</td><td>2</td></tr>
<tr><td>10260</td><td>4.0</td><td>These strings are really quite good, but I wou...</td><td>pos</td><td>2</td></tr>
</table>
10261 rows × 4 columns


</td>
<td>
Display the DataFrame with Sentiment Data
df_feedback
<table>
<tr><th></th><th>overall</th><th>feedbackText</th><th>sentiment</th><th>sentimentScore</th></tr>
<tr><td>0</td><td>5.0</td><td>Not much to write about here, but it does exac...</td><td>neg</td><td>0</td></tr>
<tr><td>1</td><td>5.0</td><td>The product does exactly as it should and is q...</td><td>pos</td><td>1</td></tr>
<tr><td>2</td><td>5.0</td><td>The primary job of this device is to block the...</td><td>pos</td><td>1</td></tr>
<tr><td>3</td><td>5.0</td><td>Nice windscreen protects my MXL mic and preven...</td><td>neg</td><td>0</td></tr>
<tr><td>4</td><td>5.0</td><td>This pop filter is great. It looks and perform...</td><td>pos</td><td>1</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>10256</td><td>5.0</td><td>Great, just as expected. Thank to all. Five S...</td><td>neg</td><td>0</td></tr>
<tr><td>10257</td><td>5.0</td><td>I've been thinking about trying the Nanoweb st...</td><td>pos</td><td>1</td></tr>
<tr><td>10258</td><td>4.0</td><td>I have tried coated strings in the past (incl...</td><td>pos</td><td>1</td></tr>
<tr><td>10259</td><td>4.0</td><td>Well, MADE by Elixir and DEVELOPED with Taylor...</td><td>pos</td><td>1</td></tr>
<tr><td>10260</td><td>4.0</td><td>These strings are really quite good, but I wou...</td><td>pos</td><td>1</td></tr>
</table>
10261 rows × 4 columns


</td>
<td>
Display the DataFrame with Sentiment Data
df_feedback
<table>
<tr><th></th><th>overall</th><th>feedbackText</th><th>sentimentScore</th></tr>
<tr><td>0</td><td>5.0</td><td>Not much to write about here, but it does exac...</td><td>2</td></tr>
<tr><td>1</td><td>5.0</td><td>The product does exactly as it should and is q...</td><td>2</td></tr>
<tr><td>2</td><td>5.0</td><td>The primary job of this device is to block the...</td><td>2</td></tr>
<tr><td>3</td><td>5.0</td><td>Nice windscreen protects my MXL mic and preven...</td><td>2</td></tr>
<tr><td>4</td><td>5.0</td><td>This pop filter is great. It looks and perform...</td><td>2</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>10256</td><td>5.0</td><td>Great, just as expected. Thank to all. Five S...</td><td>2</td></tr>
<tr><td>10257</td><td>5.0</td><td>I've been thinking about trying the Nanoweb st...</td><td>2</td></tr>
<tr><td>10258</td><td>4.0</td><td>I have tried coated strings in the past (incl...</td><td>2</td></tr>
<tr><td>10259</td><td>4.0</td><td>Well, MADE by Elixir and DEVELOPED with Taylor...</td><td>2</td></tr>
<tr><td>10260</td><td>4.0</td><td>These strings are really quite good, but I wou...</td><td>2</td></tr>
</table>
10261 rows × 3 columns




</td>
</tr>
<tr>
<td><b>There are 92% positive reviews and only 6.65% negative reviews.</b></td>
<td><b>There are 83% positive reviews and 17% negative and neutral reviews.</b></td>
<td><b>There are 88% positive reviews and only 4.5% negative reviews.</b></td>
</tr>
</table>

Model Selection and Evaluation

Raw data was made ready for modeling by using below methods:

- The free text field “feedbackText” was further processed by text cleaning:
    - Made text lowercase
    - Removed text in square brackets
    - Removed links
    - Removed special characters
    - Removed words containing numbers
    - Removed punctuation
- Split the data into 80% - Trained Data and 20% Test Data
- Applied TFIDF Vectorizer to transform the text into feature vectors that can be used as input to estimator

Model Selection

Below list of models were selected to run on the trained data

1. Multinomial Naive Bayes
2. Random Forest
3. Linear SVC
4. Logistic Regression
5. XGB Classifier with Hyperparameter tuning

With around 88% of positive customer feedback reviews, the data set was highly imbalanced.

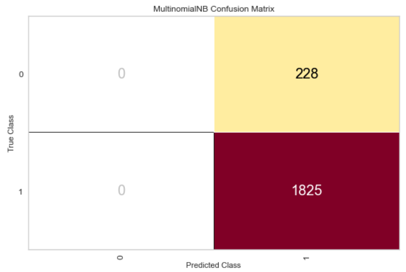
The imbalanced dataset was further managed with help of evaluating below additional methods:

6. Logistic Regression Model using class_weight
7. Neural Network Classifier
8. Neural Network Classifier with Keras
9. Logistic Regression Model - with Downsampling
10. Compare f1 Scores

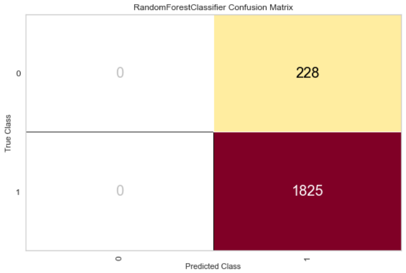
Model Evaluation

Below is the Confusion Matrix derived after running each selected model

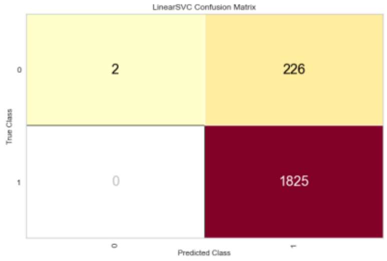
Multinomial Naive Bayes



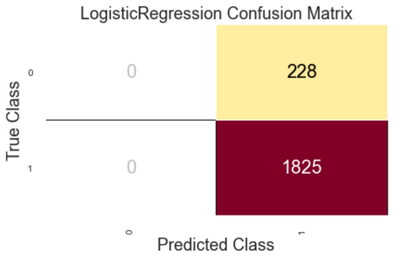
Random Forest



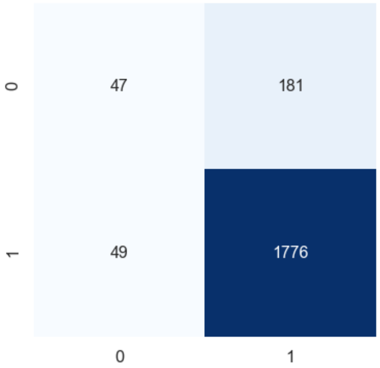
Linear SVC



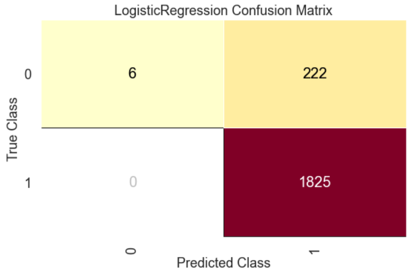
Logistic Regression



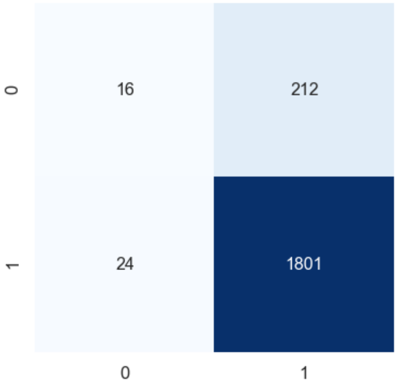
XGB Classifier



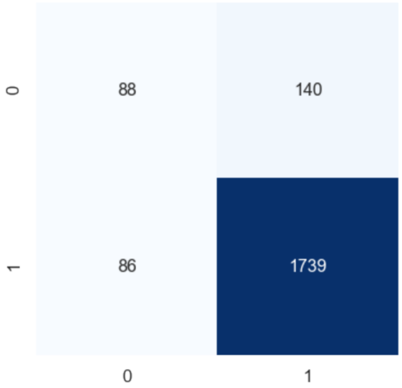
Logistic Regression (class_weight)



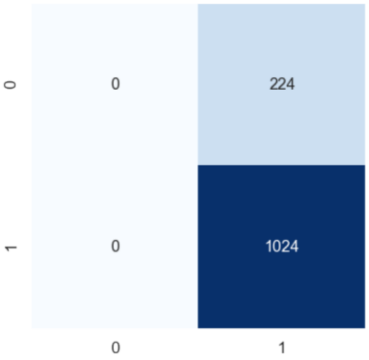
Neural Network Classifier



Neural Network Classifier – Keras



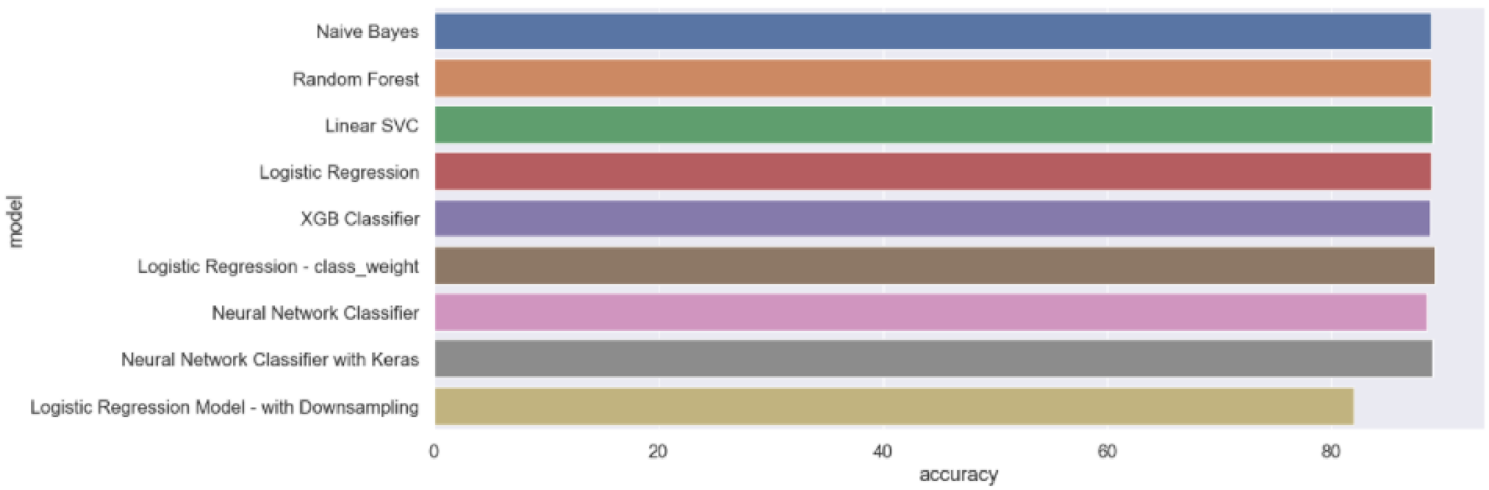
Logistic Regression Downsampling



Below is the Performance Metric of each selected model

Model	Accuracy (%)	f1 – Score (weighted avg)	Precision (weighted avg)	Recall (weighted avg)
Multinomial Naive Bayes	88.89	0.836679	0.79	0.89
Random Forest	88.89	0.836679	0.79	0.89
Linear SVC	88.99	0.839042	0.90	0.89
Logistic Regression	88.89	0.836679	0.79	0.89
XGB Classifier	88.79	0.867103	0.86	0.89
Logistic Regression (class_weight)	89.18	0.843671	0.90	0.89
Neural Network Classifier	88.50	0.847542	0.84	0.89
Neural Network Classifier - Keras	88.99	0.883326	0.88	0.89
Logistic Regression Downsampling	82.05	0.739617	0.67	0.82

Plot the Accuracy of each model based on the results above

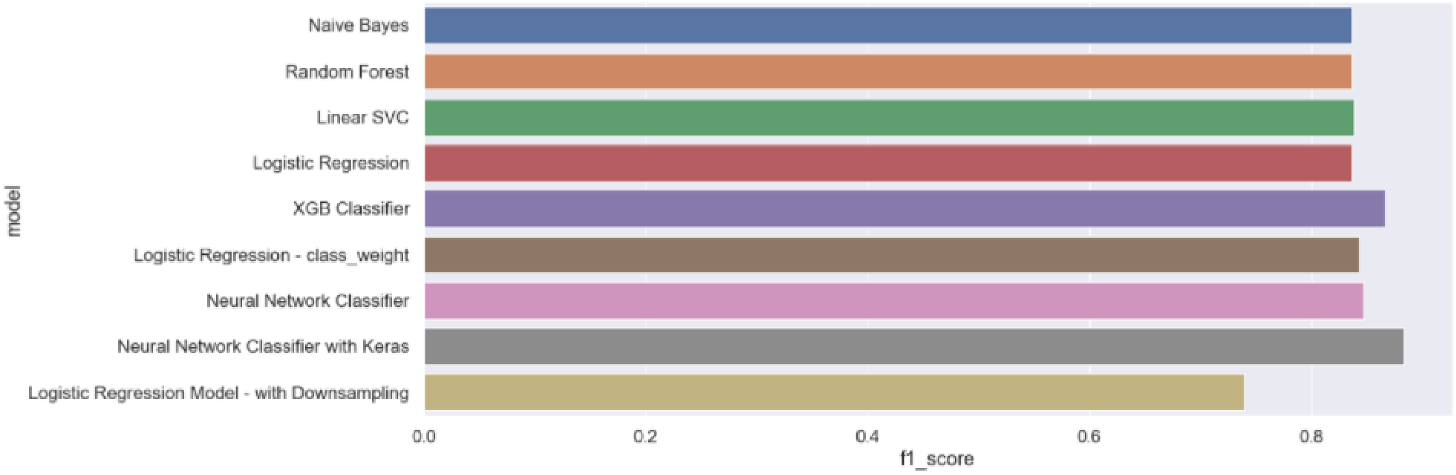


It is evident from the accuracy vs model plot above, that accuracy of imbalanced dataset is often misleading. In other words - For imbalanced data, Accuracy may not be the best measure of evaluating the model.

At the same time, However, downsampling the majority class data does show the variation in the accuracy.

f1 Scores Comparison

Plot the f1 Score of each model based on the Performance Metric above



Based on the f1 score of the weighted averages, Neural Network with Keras shows best results.

Conclusion

We have successfully processed the text data within feedbackText free text field that represents the customer feedback reviews of Musical Instruments on amazon.com. Based on the text analysis and cleaning methods followed by the Sentiment Analysis.

Challenges

We observed that we had lot of positive polarities compared to the negative polarities mainly due to large number of 5 overall ratings of the customer feedback. Our target feature contained lot of positive sentiments compared to negative and neutral. As a result, the available dataset was highly imbalanced.

It was crucial to balance the classes in such situation by alternative methods of running additional algorithms (Neural Network Classifier and Keras) as well as downsampling the dataset.

Modeling

Yet, as next step, from the sparse matrix, we predicted the classes in target feature - Overall Rating by following the selection process of multiple classification algorithms. Due to highly imbalanced dataset f1 Score's comparisons were preferred over accuracy of the model while choosing the best performing model of Neural Network Classifier with Keras.

Summary

Overall, we have followed very methodical approach by classifying all the classes starting from splitting the sentiments based on overall score, text cleaning and vectorization based on requirements and finally handling imbalance with downsampling, class_weight parameter and introducing additional models.

We would recommend the approach taken in this project that does pretty neat job in evaluating the customer feedback reviews of Musical Instruments on amazon.com to solve any similar business problem within any industry.