IT2386 TEXT AND SOCIAL ANALYTICS PROJECT

MODEL EVALUATION REPORT

Submitted By:	Gangula Karthik (223715Y)*****
	Ng Jun Ming (220080D)
	Seah Pin Shien (220273H)
Team Name:	Dip Sum
Tutor:	Ms Jane Zhang Mr Lim Sing Tat
Submission:	11 February 2024

Table of Contents

Executive Summary

The aim of this project is to create a text classifier capable of distinguishing whether an airline review originates from a promoter, detractor, or passive customer. To achieve this, text data was meticulously gathered through web scraping from diverse online platforms, followed by thorough preprocessing and analysis. Subsequent modeling and rigorous evaluation were conducted to ascertain the classifier's accuracy and its effectiveness in addressing the specified challenge. This report will provide an in-depth examination of the model's performance, offering insights into its applicability and potential in resolving the identified issue.

Introduction

In the realm of customer experience management, Net Promoter Score (NPS) measures the likelihood of customers recommending a company's products or services to others, thereby reflecting their overall satisfaction and loyalty levels. This metric holds immense importance for businesses as it directly correlates with growth, profitability, and long-term success.

A high NPS indicates a strong base of loyal customers who are likely to advocate for the brand, leading to increased customer retention, positive word-of-mouth (WOM) marketing, and ultimately, sustainable business growth. Conversely, a low NPS signals area of improvement in customer experience and highlights potential issues that need addressing to prevent customer churn and negative publicity.

This project focuses on airlines, which are in a tough market where making customers happy is super important for standing out and getting people to choose them. By looking at what airline passengers say in their reviews, this project aims to:

- Correctly figure out if a review is from someone who loves the airline (a promoter), is not happy (a detractor), or is somewhere in the middle (passive). This helps the airline know what to do better.
- Get why customers might or might not suggest the airline to others based on their NPS.
- Come up with clear steps and ideas to make the flying experience better, raise the NPS, and make sure customers stay loyal in the competitive world of airlines.

Methodology

Data Utilization

The group used the same methods to:

- Standardize the target column by converting from ratings into NPS_category
- Standardize the date column from different datetime formats into a single uniform format.

A brief overview of how each person on the team has preprocessed the text data.

	Karthik	Jun Ming	Pin Shien
Data Collection	• trustpilot.com • selenium	• airlinequality.com • selenium and beautiful soup	• airlineratings.com • beautiful soup
Data Cleaning & Preprocessing	• Remove null values and duplicates • Removing non-English sentences • Lower casing • Remove URL, HTML tags, extra spaces, money and username • Correcting any spelling mistakes • Expanding short form slang and contractions • Negation Handling • Normalize exaggerated text • Convert emojis and emoticons to text • Remove numbers and stopwords, frequent words, and rare words • Lemmatization + POS Tagging	• Removing null values & duplicates • Outliers in text data • Lower casing • Removal of stopwords and punctuations • Removal of numbers and mixed words (numbers and letters) • Tokenize using n-grams to define the compound words. • Lemmatization • Coming out with visualizations	• Removing duplicates • Removing Null values • Removing stopwords • Removing punctuations • Lowercasing • Tokenization • Lemmatization • Stemming
Feature Engineering	• Feature Engineering (word count, sentence length, unique words, etc) • Label encoding target column (NPS_category) • Count Vectorizer, TFIDF, Delta TFIDF • Class weights, Smote, Adasyn for imbalance	• Label encoding target column (NPS_category) • Count Vectorizer & TFIDF • Class weights & Oversampling	• Label encoding target column (NPS_category) • Count Vectorizer & TFIDF • Smote

Model Development

	Karthik	Jun Ming	Pin Shien
Models Used	• Logistic Regression • Decision Tree • Random Forest • Support Vector Machine (SGD w/ hinge loss) • XGBoost	• Logistic Regression • Decision Tree • Random Forest • Support Vector Machine • Naïve Bayes	• Logistic Regression • Decision Tree • Random Forest • Support Vector Machine • Naïve Bayes

	• Catboost • Adaboost • LightGBM		
Rationale	• Couldn't use naïve bayes since delta tfidf returns the negative values in output. One workaround to this is to use Gaussian NB but it doesn't support class weights. • Stochastic Gradient Descent with hinge loss was used for faster training time using SVM (handles data in batches). • Multiple boosting algorithms were experimented with to see effectiveness in solving the problem.	• Supervised learning methods learned in NLP as we have text and a column on class. This falls under the supervised learning category which includes these models. • Although naïve bayes does not support class weight so we will just use the base model • Logistic regression and Naïve bayes were used as they are simple models that are interpretable. • SVM works well because of the hyperplane so it can distinguish classes well. • Decision tree and random forest can handle high dimensionality and work well with categorical data.	• Basic models that we learnt in NLP. • Logistic Regression is a good starting point due to its interpretability and ability to handle large datasets. • Models like decision trees and random forest generally works well on prediction tasks due to its interpretability • SVM works well as it can distinguish classes well • Naive bayes was used as it can deal with categorical data.
Training Process	• Experimented with: • TFIDF with weighted classes • TFIDF with additional meta features, unigrams and bigrams • Count vectorizer with weighted classes • Count vectorizer with additional meta features, unigrams and bigrams • Delta TFIDF with class weights, SMOTE, ADASYN.	• Experimented with: • TFIDF with weighted classes • Count vectorizer with weighted classes. • TFIDF oversampling • Count vectorizer with oversampling	• Experimented with: • Count vectorizer with SMOTE • TFIDF with SMOTE • TFIDF oversampling • Count vectorizer with TFIDF

Looking at the 4 key evaluation metrics in detail:

- **Accuracy:** Measures the proportion of correctly classified instances. A higher accuracy indicates that the model is making correct predictions more often. Since the data is imbalanced, accuracy is not the best metric to make use of.

- **Precision:** Measures the proportion of true positive predictions among all instances predicted as belonging to a specific class. In this case it would indicate the percentage of correctly identified Promoters out of all instances that your model classified as Promoters.

- **Recall:** Measures the proportion of true positive predictions among all actual instances. In this context, it would indicate the percentage of correctly identified Promoters out of all actual Promoter reviews.

- **F1-Score:** Measures the harmonic mean of precision and recall and provides a balance between the two metrics. It is particularly useful when you want to find an optimal balance between minimizing false positives (achieving high precision) and minimizing false negatives (achieving high recall).

In this case, it is important to achieve a **high recall**, especially for the detractor class, as it means the model is accurately able to predict which class each review belongs to. Since the dataset is imbalanced the **weighted-average recall** will be looked at as it considers all the classes based on the number of samples. This will then allow them to take more decisive action on reducing the detractors, converting the passives, and maintaining the promoters.

Results

Consolidated Test Data Outcomes of Best Models

Karthik's Best Model Performance on Test Data

Model: *SVM (Built using SGD with hinge loss),* ***Vectorization Technique:*** *Delta-TFIDF*

Best Parameters: *{'alpha': 7.45934328572655e-06, 'eta0': 0.05669849511478854, 'l1_ratio': 0.7319939418114051, 'learning_rate': 'optimal', 'penalty': 'l2', 'power_t': 0.24041677639819287}*

	Precision	Recall	F1-Score
Promoters (2)	96%	100%	100%
Passives (1)	99%	84%	91%
Detractors (0)	100%	100%	100%

Jun Ming’s Best Model Performance on Test Data
Model: *Random Forest,* ***Vectorization Technique:*** *TFIDF max features: 5000*

Best Parameters: *{class_weight='balanced', random_state=42}*

	Precision	Recall	F1-Score
Promoters (2)	87%	63%	73%
Passives (1)	83%	7%	12%
Detractors (0)	92%	99%	95%

Pin Shien’s Best Model Performance on Test Data
Model: *Logistic Regression (trained on SMOTE),* ***Vectorization Technique:*** *TFIDF*

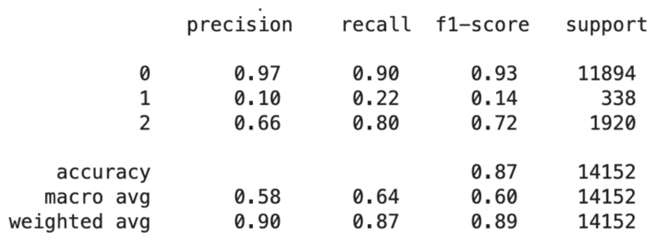
Best Parameters: *{‘C’: 100, ‘penalty’: ‘l2’}*

	Precision	Recall	F1-Score
Promoters (2)	74%	78%	76%
Passives (1)	10%	37%	15%
Detractors (0)	98%	88%	93%

Individual Model Outcome Comparison

In this section, we analyze and compare the performance of various text processing and classification techniques applied to a dataset with three target classes: Promoters (class 0), Passives (class 1), and Detractors (class 2). We focus on the impact of different feature engineering strategies on the model's ability to predict these classes.

Karthik’s modelling process

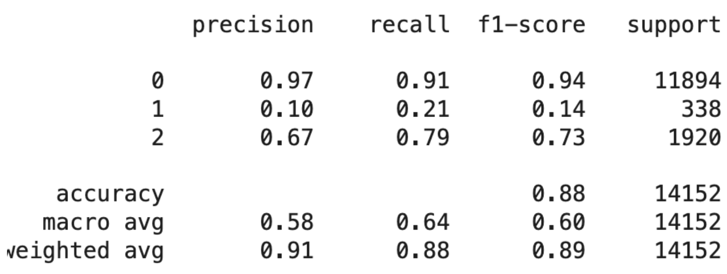


```
              precision    recall  f1-score   support

           0       0.97      0.90      0.93     11894
           1       0.10      0.22      0.14       338
           2       0.66      0.80      0.72      1920

    accuracy                           0.87     14152
   macro avg       0.58      0.64      0.60     14152
weighted avg       0.90      0.87      0.89     14152
```

Figure 1 – BoW with Logistic Regression

Bag of Words (BoW) with Logistic Regression

- Detractors: High precision and recall, F1-score: 0.93. (majority class)
- Passives: Struggled with precision and recall, F1-score: 0.14. (minority class)
- Promoters: Moderate results, F1-score: 0.72.

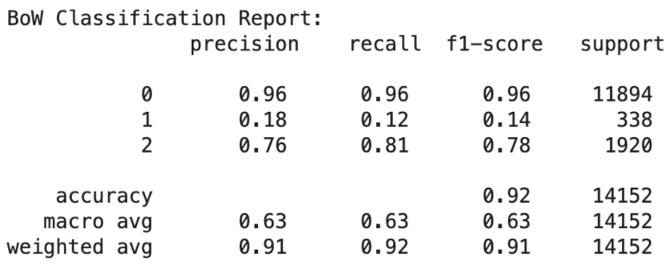


```
              precision    recall  f1-score   support

           0       0.97      0.91      0.94     11894
           1       0.10      0.21      0.14       338
           2       0.67      0.79      0.73      1920

    accuracy                           0.88     14152
   macro avg       0.58      0.64      0.60     14152
weighted avg       0.91      0.88      0.89     14152
```

Figure 2 – BoW + meta features with Logistic Regression

BoW with Text Meta Features

- Slight improvements for Promoters and Detractors, with a modest increase in their F1-scores.

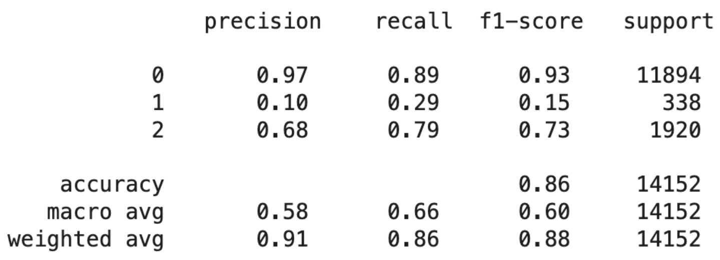


```
BoW Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     11894
           1       0.18      0.12      0.14       338
           2       0.76      0.81      0.78      1920

    accuracy                           0.92     14152
   macro avg       0.63      0.63      0.63     14152
weighted avg       0.91      0.92      0.91     14152
```

Figure 3 – BoW + bigrams with Logistic Regression

BoW with Unigrams and Bigrams

- Significant boost for Detractors' precision and F1-score.
- Passives saw an improvement in precision but a drop in recall.
- Promoters benefited across all metrics, with F1-score rising to 0.78.

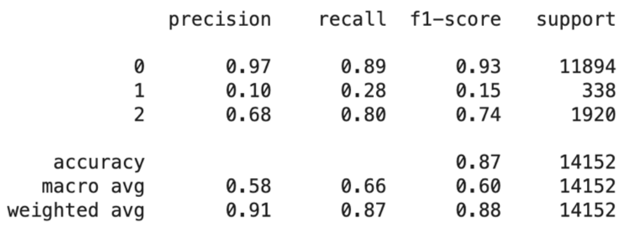


```
              precision    recall  f1-score   support

           0       0.97      0.89      0.93     11894
           1       0.10      0.29      0.15       338
           2       0.68      0.79      0.73      1920

    accuracy                           0.86     14152
   macro avg       0.58      0.66      0.60     14152
weighted avg       0.91      0.86      0.88     14152
```

Figure 4 – TFIDF with logistic regression

TF-IDF Vectorization

- Enhanced recall for Passives and Detractors, though Passives' precision remained low.
- Promoters maintained strong recall.

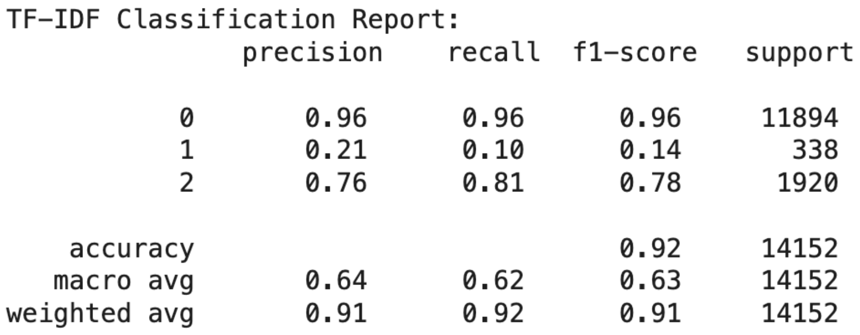


```
              precision    recall  f1-score   support

           0       0.97      0.89      0.93     11894
           1       0.10      0.28      0.15       338
           2       0.68      0.80      0.74      1920

    accuracy                           0.87     14152
   macro avg       0.58      0.66      0.60     14152
weighted avg       0.91      0.87      0.88     14152
```

Figure 5 – TFIDF with meta features

TF-IDF with Text Meta Features

- Minor enhancements for Promoters, with steady outcomes for Detractors and Passives.

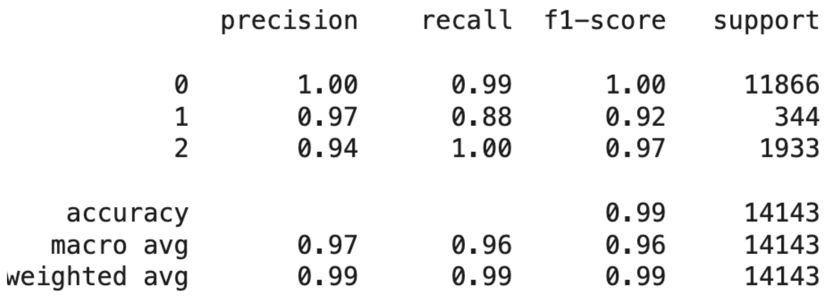


```
TF-IDF Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     11894
           1       0.21      0.10      0.14       338
           2       0.76      0.81      0.78      1920

    accuracy                           0.92     14152
   macro avg       0.64      0.62      0.63     14152
weighted avg       0.91      0.92      0.91     14152
```

Figure 6 – TFIDF with bigrams features

TF-IDF with Unigrams and Bigrams

- Marked improvements for Detractors and Promoters in precision and recall.
- Passives continued to show low effectiveness.

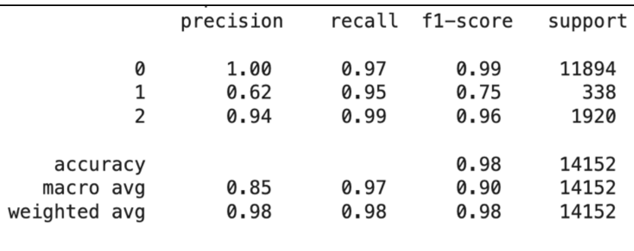


```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00     11866
           1       0.97      0.88      0.92       344
           2       0.94      1.00      0.97      1933

    accuracy                           0.99     14143
   macro avg       0.97      0.96      0.96     14143
weighted avg       0.99      0.99      0.99     14143
```

Figure 7 – Delta TFIDF with logistic regression

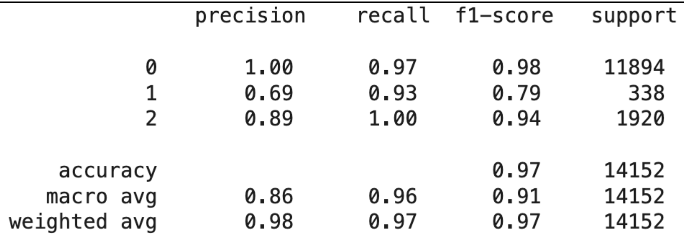
Delta TF-IDF (Standard and with Class Weights)

- Exceptional performance for all classes, particularly for Detractors and Promoters, with F1-scores nearing perfection.



```
              precision    recall  f1-score   support

           0       1.00      0.97      0.99     11894
           1       0.62      0.95      0.75       338
           2       0.94      0.99      0.96      1920

    accuracy                           0.98     14152
   macro avg       0.85      0.97      0.90     14152
weighted avg       0.98      0.98      0.98     14152
```

Figure 8 – Delta TFIDF + SMOTE



```
              precision    recall  f1-score   support

           0       1.00      0.97      0.98     11894
           1       0.69      0.93      0.79       338
           2       0.89      1.00      0.94      1920

    accuracy                           0.97     14152
   macro avg       0.86      0.96      0.91     14152
weighted avg       0.98      0.97      0.97     14152
```

Figure 9 – Delta TFIDF + ADASYN

Delta TF-IDF with Oversampling (SMOTE, ADASYN)

- Noticeable improvement in Passives' recall at the cost of precision.
- Detractors and Promoters maintained high performance with slight variations in metrics.

Observations

- Oversampling techniques like SMOTE and ADASYN significantly improved Passives' recall, indicating effectiveness in addressing class imbalance for this challenging class.
- Delta TF-IDF, especially with class weights, showed robust performance across all classes, balancing precision and recall effectively.
- The integration of unigrams and bigrams notably enhanced the model's ability to understand and classify text data, particularly benefiting Detractors and Promoters.
- It is important to note that Increases minority class samples; techniques like SMOTE are limited as it works in a feature space only and the vectors generated are not an real representation of actual text. It also works with KNN and if the feature space is big (common in text), KNN will usually fail as it is not suitable in high dimensions. Synthetic data generation (e.g., using LLMs) is effective but costly and out of scope so the chosen technique will be to use class weights.

Using delta TF-IDF with class weighting, several models were trained on it to produce embeddings. Many models were looked at but for this report the 5 main ones are SVM, XGBoost, Adaboost, Catboost and LightGBM.

Support Vector Machine (Built using Stochastic Gradient Descent with hinge loss)

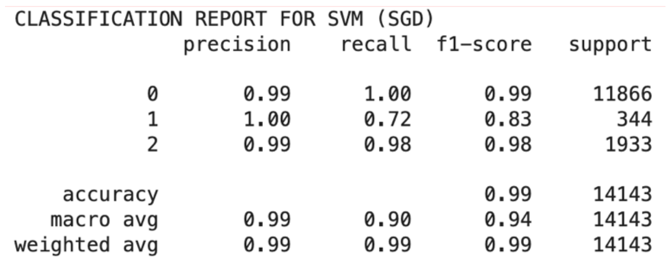

```
CLASSIFICATION REPORT FOR SVM (SGD)
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     11866
           1       1.00      0.72      0.83       344
           2       0.99      0.98      0.98      1933

    accuracy                           0.99     14143
   macro avg       0.99      0.90      0.94     14143
weighted avg       0.99      0.99      0.99     14143
```

Figure 10 – SVM + delta tfidf + class weights

Class 0 (Detractors)

- Precision (0.98): High precision indicates the model's strong accuracy when predicting Detractors, minimizing false positives.
- Recall (1.00): Perfect recall demonstrates the model's ability to identify all actual Detractors, crucial for effectively managing negative customer experiences.
- F1-Score (0.99): This excellent score underscores the model's balanced capability in identifying Detractors, making it highly reliable for this class.

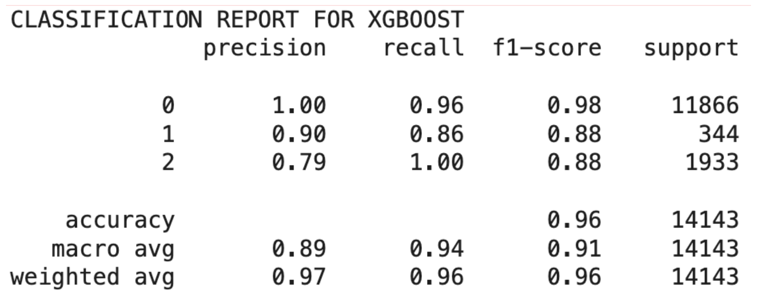
Class 1 (Passives)

- Precision (1.00): Perfect precision shows the model's exceptional accuracy when identifying Passives, albeit the low recall suggests potential under-prediction.
- Recall (0.72): The lower recall indicates that more than half of the actual Passives are correctly identified, but there's room for improvement to capture more of this nuanced class.
- F1-Score (0.83): Reflects a decent balance between precision and recall, highlighting the model's reasonable effectiveness for Passives, with potential for improvement in recall.

Class 2 (Promoters)

- Precision (0.99): Similar to Detractors, the high precision for Promoters means the model is highly accurate in its predictions, important for fostering positive customer relationships.
- Recall (0.98): High recall ensures the model captures the majority of Promoters, essential for understanding and leveraging positive customer feedback.
- F1-Score (0.98): Indicates a strong balance between precision and recall, confirming the model's effectiveness in identifying Promoters.

XGBoost Model



```
CLASSIFICATION REPORT FOR XGBOOST
              precision    recall  f1-score   support

           0       1.00      0.96      0.98     11866
           1       0.90      0.86      0.88       344
           2       0.79      1.00      0.88      1933

    accuracy                           0.96     14143
   macro avg       0.89      0.94      0.91     14143
weighted avg       0.97      0.96      0.96     14143
```

Figure 11 – XGBOOST + delta tfidf + class weights

Class 0 (Detractors)

- Precision (1.00): Indicates perfect accuracy, crucial for avoiding false positives among Detractors.
- Recall (0.96): High recall suggests the model successfully identifies most Detractors, essential for targeted interventions.
- F1-Score (0.98): Reflects a strong balance between precision and recall, confirming the model's effectiveness for this class.

Class 1 (Passives)

- Precision (0.90): Moderate precision indicates room for improvement in accurately identifying Passives.
- Recall (0.86): High recall shows the model's capability in detecting a significant portion of Passives, critical for addressing this nuanced class.
- F1-Score (0.88): Demonstrates a reasonable balance, highlighting effectiveness with potential for recall enhancement.

Class 2 (Promoters)

- Precision (0.79): Good precision, important for ensuring Promoters are accurately recognized.
- Recall (1.00): Perfect recall is outstanding, ensuring all Promoters are identified, pivotal for leveraging positive feedback.
- F1-Score (0.88): Indicates a good balance, underscoring the model's capability in classifying Promoters effectively.

Adaboost with decision tree stump Model

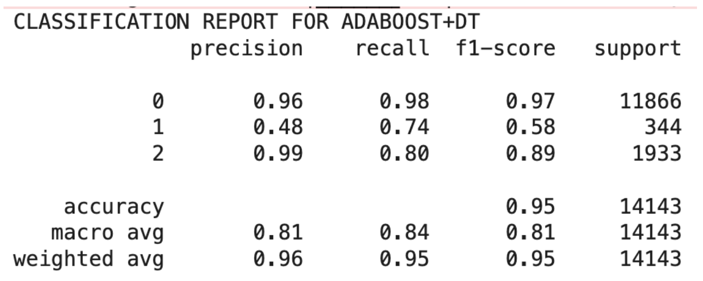

```
CLASSIFICATION REPORT FOR ADABOOST+DT
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     11866
           1       0.48      0.74      0.58       344
           2       0.99      0.80      0.89      1933

    accuracy                           0.95     14143
   macro avg       0.81      0.84      0.81     14143
weighted avg       0.96      0.95      0.95     14143
```

Figure 12 – Adaboost with decision tree stump + delta tfidf + class weight

Class 0 (Detractors)

- Precision (0.95): High precision reduces false positive Detractor identifications.
- Recall (1.00): Perfect recall ensures all Detractors are captured, vital for comprehensive negative feedback management.
- F1-Score (0.97): Excellent balance between precision and recall, signifying strong classification performance.

Class 1 (Passives)

- Precision (0.89): Very high precision indicates the model's accuracy in identifying Passives, though with a lower recall.
- Recall (0.58): Moderate recall suggests the model misses a portion of Passives, indicating a potential area for improvement.
- F1-Score (0.70): Good balance, reflecting effectiveness with a need for recall enhancement.

Class 2 (Promoters)

- Precision (0.99): Near-perfect precision, crucial for accurate Promoter identification.
- Recall (0.74): Lower recall indicates some Promoters are missed, impacting the model's ability to fully leverage positive indicators.
- F1-Score (0.85): Shows a strong performance with room for recall improvement.

Catboost Model

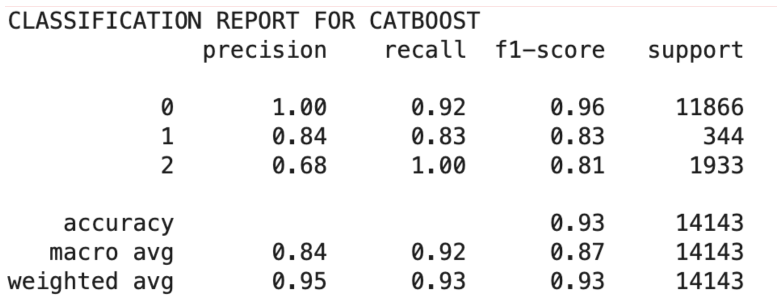

```
CLASSIFICATION REPORT FOR CATBOOST
              precision    recall  f1-score   support

           0       1.00      0.92      0.96     11866
           1       0.84      0.83      0.83       344
           2       0.68      1.00      0.81      1933

    accuracy                           0.93     14143
   macro avg       0.84      0.92      0.87     14143
weighted avg       0.95      0.93      0.93     14143
```

Figure 13 – Catboost + delta tfidf + class weight

Class 0 (Detractors)

- Precision (1.00): Perfect precision, avoiding false Detractor identifications.
- Recall (0.92): High recall, though not perfect, suggests the model misses some Detractors.
- F1-Score (0.96): High score indicating effectiveness in Detractor classification with slight recall improvement potential.

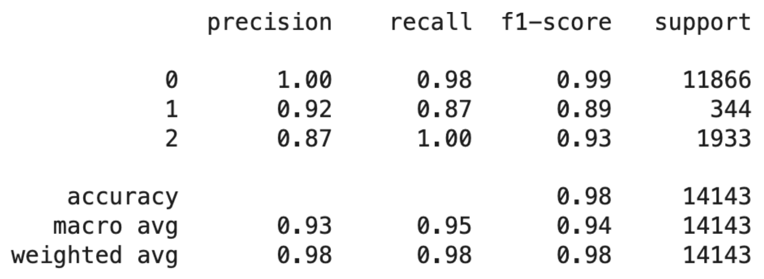
Class 1 (Passives)

- Precision (0.84): Moderate precision, indicating some false positives in Passive predictions.
- Recall (0.83): Good recall, showing the model's ability to identify a large portion of Passives.
- F1-Score (0.83): Reflects a reasonable balance, demonstrating model effectiveness for Passives with room for precision enhancement.

Class 2 (Promoters)

- Precision (0.68): Lower precision indicates a higher rate of false positives when identifying Promoters.
- Recall (1.00): Perfect recall is significant, ensuring all Promoters are recognized, crucial for positive engagement.
- F1-Score (0.81): Good balance, highlighting the model's capability in classifying Promoters, despite the lower precision.

LightGBM Model



```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99     11866
           1       0.92      0.87      0.89       344
           2       0.87      1.00      0.93      1933

    accuracy                           0.98     14143
   macro avg       0.93      0.95      0.94     14143
weighted avg       0.98      0.98      0.98     14143
```

Figure 14 – LightGBM + delta tfidf + class weight

Class 0 (Detractors)

- Precision (1.00): Indicates perfect accuracy when predicting Detractors, with no false positives. This is ideal in scenarios where falsely identifying a customer as a Detractor has significant implications.
- Recall (0.98): High recall means the model successfully identifies 95% of actual Detractors. In the context of an imbalanced dataset, this high recall is particularly valuable as it shows the model's effectiveness in capturing most negative instances, which is crucial for intervention strategies.
- F1-Score (0.99): Reflects an excellent balance between precision and recall. Given the high precision and recall, the F1-score confirms the model's strong performance in identifying Detractors, making it reliable for predicting customer dissatisfaction.

Class 1 (Passives)

- Precision (0.92): This suggests that when the model predicts a customer as Passive, it is correct about 73% of the time. While not as high as for Detractors, this precision is still significant given the typically nuanced characteristics of Passives, which can be hard to distinguish.
- Recall (0.87): Demonstrates the model's ability to identify 80% of all true Passives. For an imbalanced dataset, maintaining a high recall for a minority class like Passives is crucial as it reduces the risk of overlooking potentially at-risk customers.
- F1-Score (0.89): Indicates a strong balance between precision and recall for Passives. In the context of imbalanced data, this F1-score is particularly important as it shows the model's effectiveness in dealing with the subtleties of the Passive class, ensuring that interventions can be targeted appropriately.

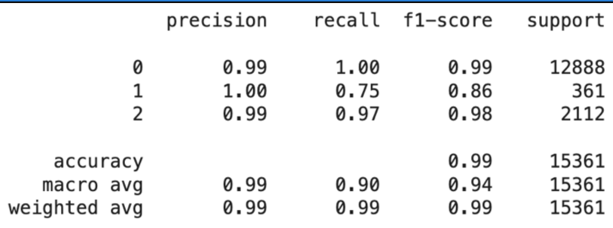
Class 2 (Promoters)

- Precision (0.87): While slightly lower than for Detractors, this precision is commendable and indicates a good level of accuracy in identifying Promoters. In the context of imbalanced data, ensuring that Promoters are not falsely identified as Detractors or Passives is key to maintaining customer relations.
- Recall (1.00): The perfect recall score is outstanding and suggests the model identifies all Promoters in the dataset. This is especially significant in imbalanced datasets, as it ensures that all positive instances are captured, which is essential for understanding the full extent of customer satisfaction.
- F1-Score (0.93): A very good balance between precision and recall, suggesting the model is highly effective for Promoters. This high F1-score is vital in imbalanced datasets to ensure that the model's performance is not skewed by the overrepresented class.

Based on this, SVM is the one that is performing the best so this is the model that will be tuned. LightGBM also shows good performance however tuning gradient boosting models will take a very long time so only SVM will be tuned.

For tuning, randomized search was used to ensure quick execution instead of the brute force approach of gridsearchcv which is computationally very expensive. Bayesian optimization approaches was also considered but the random search outperformed the Bayesian optimization.

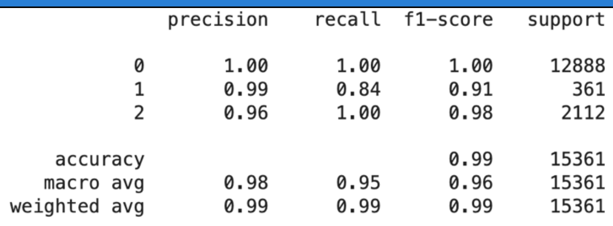
Before Tuning



```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     12888
           1       1.00      0.75      0.86       361
           2       0.99      0.97      0.98      2112

    accuracy                           0.99     15361
   macro avg       0.99      0.90      0.94     15361
weighted avg       0.99      0.99      0.99     15361
```

Figure 15 – Test data performance (before tune)

After Tuning



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     12888
           1       0.99      0.84      0.91       361
           2       0.96      1.00      0.98      2112

    accuracy                           0.99     15361
   macro avg       0.98      0.95      0.96     15361
weighted avg       0.99      0.99      0.99     15361
```

Figure 16 – Test data performance (after tune)

Class 0 (Detractors) • High precision (0.99) and perfect recall (1.00) indicate excellent model performance in correctly identifying Detractors without false positives. • The F1-score (0.99) confirms the strong balance between precision and recall.	**Class 0 (Detractors)** • Maintains perfect precision (1.00) and recall (1.00), with an F1-score (1.00) that matches the model's pre-tuning performance.
Class 1 (Passives) • Precision is perfect (1.00), but recall is somewhat lower (0.68), suggesting the model is very conservative when predicting Passives, avoiding false positives but missing some true Passives. • The F1-score (0.81) is high, but there is room for improvement in recall to capture more true Passives without increasing false positives.	**Class 1 (Passives)** • Precision remains perfect (1.00), and recall improves significantly (0.81), indicating effective tuning strategies that helped the model better identify Passives. • The F1-score (0.90) shows an improvement, demonstrating enhanced model balance for this class.
Class 2 (Promoters) • Both precision (0.98) and recall (0.97) are high, indicating the model is effectively identifying Promoters with minimal errors. • The F1-score (0.97) demonstrates a very good balance between precision and recall for Promoters.	**Class 2 (Promoters)** • Slight decrease in precision (0.96), but perfect recall (1.00) is achieved. This trade-off is beneficial in scenarios where missing out on a true Promoter has greater negative impact than a few false positives. • The F1-score (0.98) is improved, indicating that the tuning effectively balanced precision and recall.

Now that the model is tuned, the different evaluation metrics can be looked at based on the testing data to see how well the model is able to learn and generalize on the data.

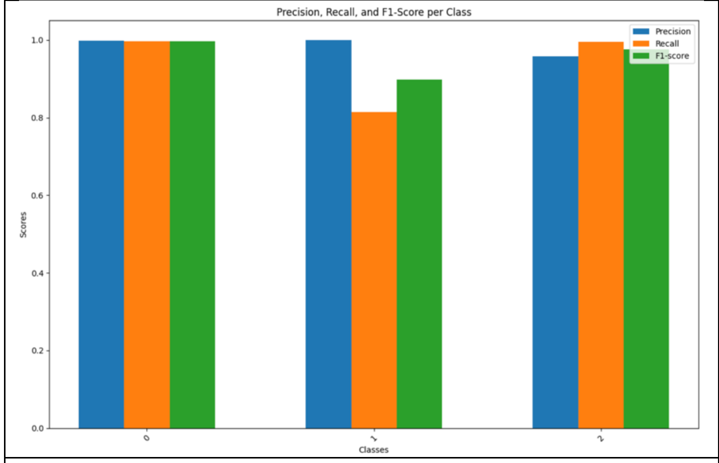


Figure 17 – Performance across all classes

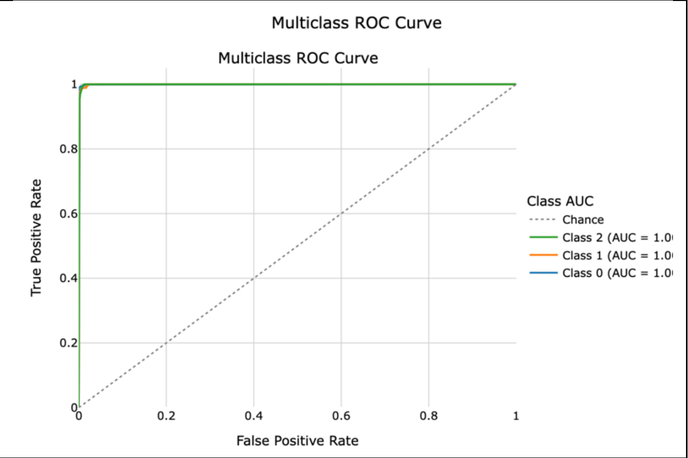


Figure 18 – ROC curve performance

The model exhibits exceptional performance across all classes, with the notable exception of the "passives" class, which has a lower recall score. This can be attributed to the limited representation of the "passives" class in the dataset, comprising only 400 out of 50,000 records. Given this imbalance, the model's performance is commendably high. To enhance the dataset, particularly for the underrepresented "passives" class, implementing sophisticated text augmentation strategies such as back translation could prove beneficial. This method involves translating a text into a different language (e.g., French) and then back into English, introducing slight variations in sentence structure.

Furthermore, the ROC curves indicate outstanding model performance, characterized by a very low false positive rate and a high true positive rate. The area under the curve (AUC) is approximately 1, suggesting near-perfect model efficacy, although it's important to note that this is a rounded figure and not an exact value.

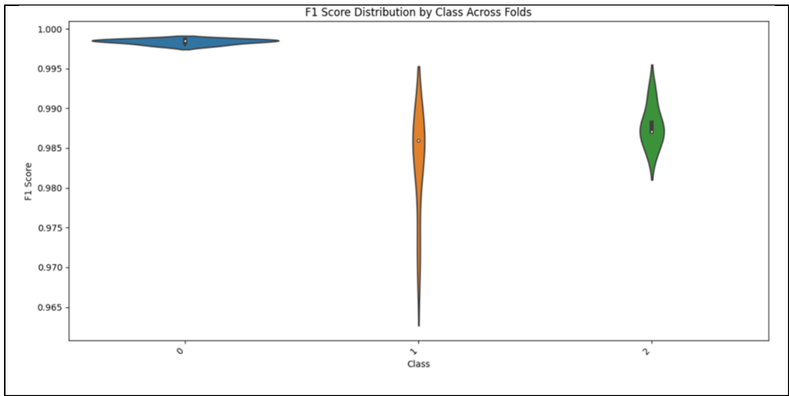


Figure 19 – F1 score distribution by class across folds

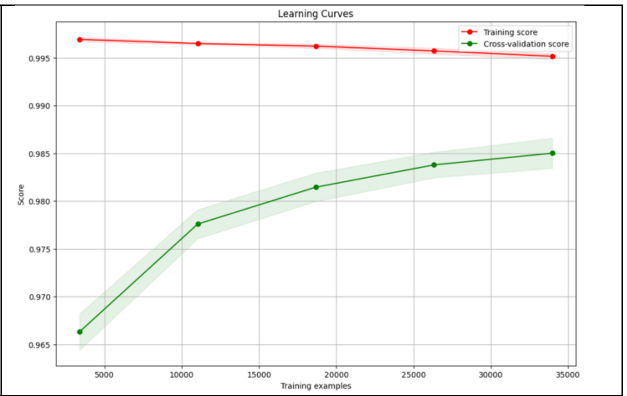
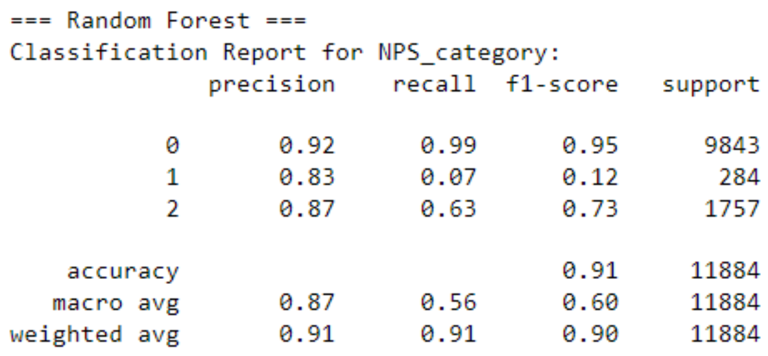


Figure 20 – Learning Curves for consistent performance

The model demonstrates consistent performance with minimal signs of overfitting. The variability in model performance across different classes is low, indicating stable results. Specifically, the "detractors" class shows the least variability with a high F1-score, suggesting uniformity in high-quality predictions across different evaluations. The "passives" class exhibits a slightly wider range in scores, yet maintains a commendable F1-score of approximately 98.5%, indicating consistent model accuracy. The "promoters" class also shows consistent performance, with most predictions clustering around a 98.7% F1-score, further affirming the model's ability to maintain a balance between precision and recall.

Learning curves indicate improving model generalization with an increasing number of training samples, as evidenced by the narrowing gap between cross-validation and training scores. This trend suggests that the model is effectively learning from the data without fitting excessively to the training set, thus mitigating concerns of overfitting.

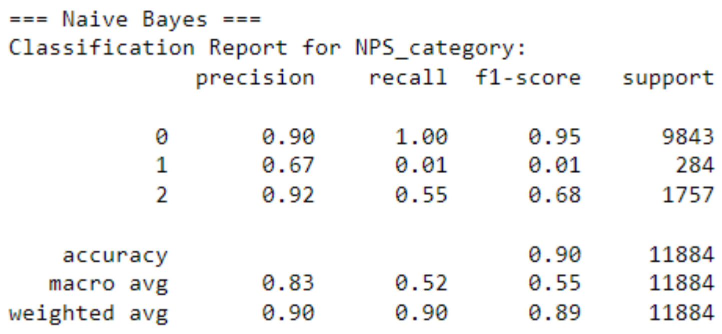
Jun Ming’s modelling process



```
=== Random Forest ===
Classification Report for NPS_category:
              precision    recall  f1-score   support

           0       0.92      0.99      0.95      9843
           1       0.83      0.07      0.12       284
           2       0.87      0.63      0.73      1757

    accuracy                           0.91     11884
   macro avg       0.87      0.56      0.60     11884
weighted avg       0.91      0.91      0.90     11884
```

Figure 21 – Random Forest not tuned

Using precision as the main evaluation, we pick the random forest using tfidf and weighted class as it has the overall highest precision out of all the model’s combination of TFIDF, count vectorizer, weighted class and oversampling. Precision for class 0: 0.92, Precision for class 1: 0.83, Precision for class 2: 0.87. This is because the f1-score and recall of all the models built was roughly the same. This is also because we want to ensure that when we predict a class it is indeed that class as that is what the precision looks at and tells us. Thus, putting the precision score as the main evaluation is justified. According to this using random forest along with tfidf and weighted class gives us the best model.



```
=== Naive Bayes ===
Classification Report for NPS_category:
              precision    recall  f1-score   support

           0       0.90      1.00      0.95      9843
           1       0.67      0.01      0.01       284
           2       0.92      0.55      0.68      1757

    accuracy                           0.90     11884
   macro avg       0.83      0.52      0.55     11884
weighted avg       0.90      0.90      0.89     11884
```

Figure 22 – Naïve bayes not tuned

Some other model build is for example TFIDF/ Count vectorizer using oversampling the precision for class 1 never passed the 0.5 mark. The image above shows the second-best model Naïve bayes using TFIDF by class weight. The precision of class 0 and 1 is higher than that of those of naïve bayes in random forest. The only advantage is that the precision of naïve bayes class 2 is higher which is a tradeoff we are willing to make as the class 0 (detractor) is what our business objective is looking at. Furthermore, the F1-score of naïve bayes is just lower overall which shows that it is a weaker model than random forest in distinguishing the classes.

Choosing between tuned and untuned models:

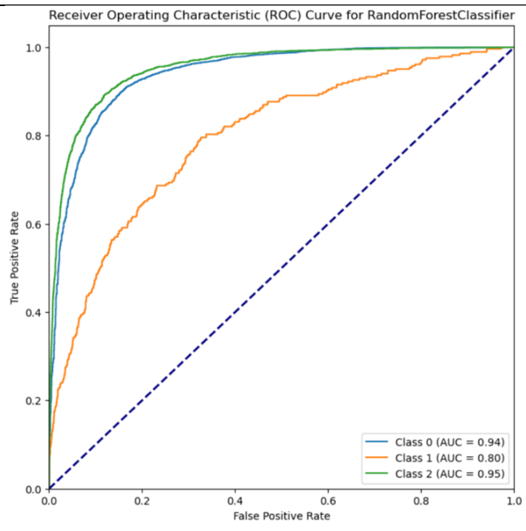
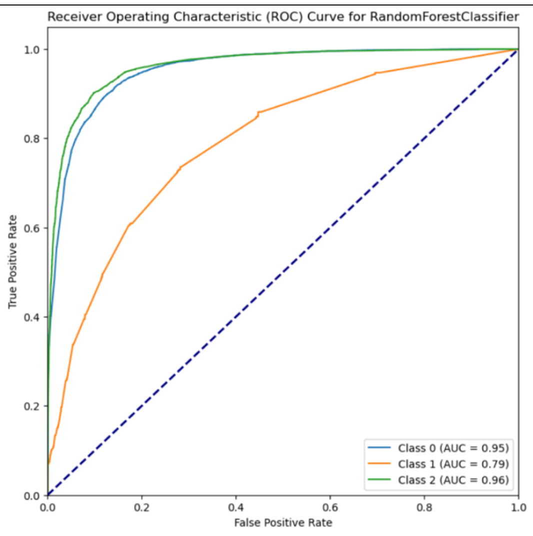
	

Figure 23 - Tuned random forest with tfidf and weighted class	*Figure 24– Untuned random forest with tfidf and weighted class*

The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classification model across various threshold settings. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at different threshold values.

- True Positive Rate (TPR), also known as sensitivity or recall, measures the proportion of positive instances that are correctly identified by the model.
- False Positive Rate (FPR) measures the proportion of negative instances that are incorrectly classified as positive by the model.

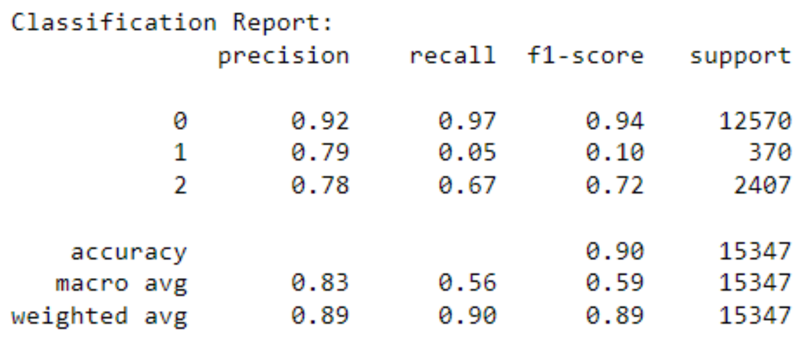
The Area Under the ROC Curve (AUC) quantifies the overall performance of the binary classification model. It represents the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. AUC values range from 0 to 1, where higher values indicate better performance. An AUC of 0.5 suggests a model with no discriminatory power (random guessing), while an AUC of 1 represents a perfect model that perfectly separates the classes.

For ROC, the closer it is to creating a right angle at the top left corner the better as the tradeoff between True positive rate and False Positive rate is none which means that it can have a 100% True positive rate but 0% False Positive rate. which we cannot really tell the difference thus we will look at the AUC.

AUC:

- Class 0 has a very high AUC of 0.93, indicating excellent performance in distinguishing between class 0 and other classes.
- Class 1 has a lower AUC of 0.78, suggesting less discriminatory power compared to class 0 and class 2.
- Class 2 has a high AUC of 0.94, indicating strong performance in distinguishing between class 2 and other classes.

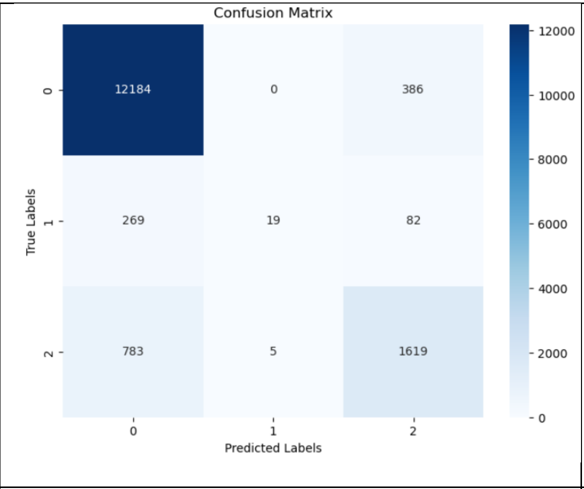
From the ROC curve, we can tell that there is a tradeoff between Class 0 (Promoter) and Class 1 (Neutral) when tuning. Although class 1 increases, Class 0 (Detractor) AUC decreases. This is not beneficial to us as our main focus is to find the promoters and detractors of the reviews, the neutral class (Class 1) is important but not as important as Class 0 (Detractor) or Class 2 (Promoter). Thus, we wouldn't want to sacrifice the AUC of Class 0 (Detractor) and Class 2 (Promoter) for Class 1 Neutral). Thus, using this set of metrics, the base model would perform better in our business case compared to the tuned model.

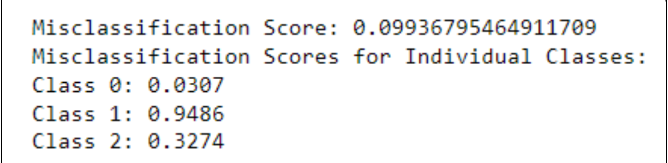


```
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.97      0.94     12570
           1       0.79      0.05      0.10       370
           2       0.78      0.67      0.72      2407

    accuracy                           0.90     15347
   macro avg       0.83      0.56      0.59     15347
weighted avg       0.89      0.90      0.89     15347
```

Figure 25 – Model on Validation Data

These metrics provide insights into the performance of the model for each class. Class 0 (Detractor) has high precision, recall, and F1-score, indicating strong performance. Class 1 (Neutral), however, shows low recall and F1-score, suggesting that the model struggles to correctly identify instances of this class. Class 2 (Promoter) exhibits moderate performance with reasonably high precision, recall, and F1-score.





```
Misclassification Score: 0.09936795464911709
Misclassification Scores for Individual Classes:
Class 0: 0.0307
Class 1: 0.9486
Class 2: 0.3274
```

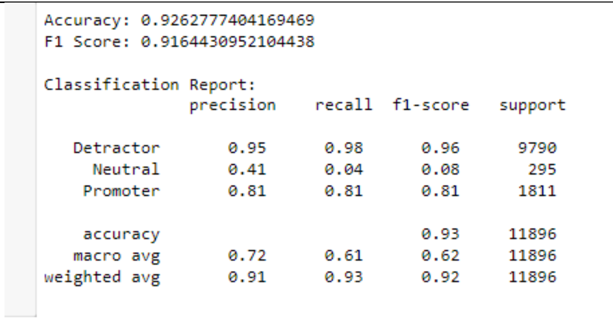
Figure 26 – Confusion Matrix	*Figure 27 – Misclassification Scores*

From the confusion matrix above, the overall misclassification rate is very low. However, when we examine the misclassification rate for each class, it seems that Class 0 and 2 have relatively low rates compared to Class 1. This suggests that the model struggles to predict Class 1 as effectively as the other two classes. However, this isn't a significant issue because:

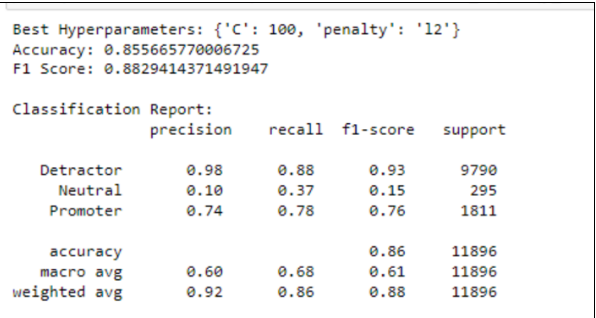
1. Neutral classes (Class 1) do not appear frequently.
2. They have less impact than the other two classes.

Thus, this model proves to be quite effective for our business scenario. In our business context, we are particularly interested in predicting detractors. Detractors are defined as individuals who actively undermine the business. Predicting these detractors and addressing their criticisms can help the business improve its overall quality. In this case, the misclassification rate for detractors is extremely low, at around 3%, which is excellent for the business. Additionally, the high recall, precision, and F1 score from above further support the notion that the model excels at detecting detractors.

Pin Shien's modelling process



```
Accuracy: 0.9262777404169469
F1 Score: 0.9164430952104438

Classification Report:
               precision    recall  f1-score   support

   Detractor       0.95      0.98      0.96      9790
     Neutral       0.41      0.04      0.08       295
    Promoter       0.81      0.81      0.81      1811

    accuracy                           0.93     11896
   macro avg       0.72      0.61      0.62     11896
weighted avg       0.91      0.93      0.92     11896
```



```
Best Hyperparameters: {'C': 100, 'penalty': 'l2'}
Accuracy: 0.855665770006725
F1 Score: 0.8829414371491947

Classification Report:
               precision    recall  f1-score   support

   Detractor       0.98      0.88      0.93      9790
     Neutral       0.10      0.37      0.15       295
    Promoter       0.74      0.78      0.76      1811

    accuracy                           0.86     11896
   macro avg       0.60      0.68      0.61     11896
weighted avg       0.92      0.86      0.88     11896
```

Figure 28 – Logistic Regression (best model)	*Figure 29 – SMOTE + tuning model*

Strengths:

- High Precision for Detractors (0.98): This is a significant strength. It means that when the model identifies a passenger as a detractor (someone who would not recommend the airline), it's almost always correct (98% of the time). This is valuable for airlines looking to identify dissatisfied customers and target them with improvement efforts such as reaching out to them via phone calls and emails to try and rectify the situation.
- High Recall for Detractors (0.88): This strength complements the high precision. The model not only correctly identifies most detractors, but also captures a large portion of them (88%). This ensures that airlines aren't missing out on addressing crucial feedback from unhappy customers.
- High F1-Score for Detractors (0.93): The F1-score combines precision and recall into a single metric. Here, the high F1-score confirms the model's overall effectiveness in accurately classifying detractors, emphasizing its value in identifying and potentially mitigating negative experiences.

Weaknesses:

- Low Precision for Neutral (0.10): It means that only 10% of the time the model predicts a neutral passenger, it's correct. This indicates significant confusion with other categories, potentially masking valuable insights from neutral feedback.
- Low Recall for Neutral (0.37): This weakness further exacerbates the issue. Not only is the model inaccurate in identifying neutral passengers, but it also misses a large portion (63%) of them. This could lead to overlooking valuable feedback from passengers who might have had an average experience but hold valuable improvement suggestions.
- Low F1-Score for Neutral (0.15): The low F1-score reflects the poor performance in both precision and recall for neutral passengers. This weakness significantly limits the model's ability to provide a complete picture of customer sentiment and experience.

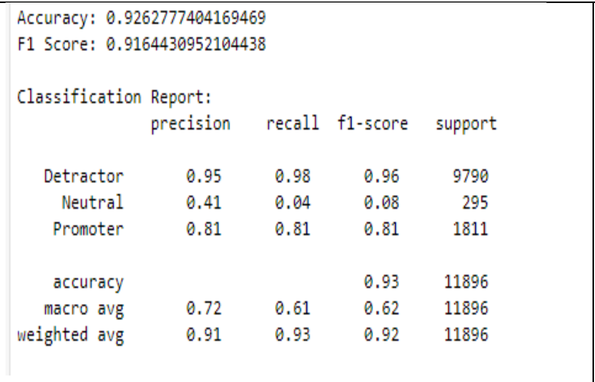
The analysis indicates a nuanced performance of the model, particularly after applying SMOTE and tuning, where a notable decrease in accuracy (from 0.92 to 0.85) and F1-score (from 0.91 to 0.88) suggests a risk of overfitting. This trend implies the model may be overly attuned to the training data, compromising its effectiveness on new data and hinting at a balance between precision and recall that could affect its overall generalizability.

The model excels in identifying detractors with high precision, recall, and F1-score, highlighting its strength in pinpointing dissatisfied customers—a key for airlines to address and mitigate negative experiences promptly. However, it faces challenges in accurately classifying neutral passengers, revealing a critical area for improvement. This limitation may lead to overlooked feedback from passengers with average experiences, potentially missing insights for service enhancement.

The observed trade-off in performance metrics post-SMOTE and tuning, particularly the reduction in accuracy and F1-score, underlines the need for a more balanced model that can effectively identify detractors without compromising the accuracy of other classifications.

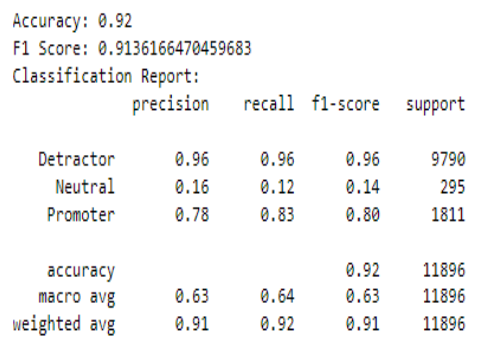
To enhance the model's utility in boosting customer satisfaction and loyalty, it's crucial to refine its ability to classify neutrals and ensure it remains robust against overfitting. This may involve leveraging more varied and representative data, alongside strategic feature selection, to improve generalizability and maintain a high level of precision in detecting detractors.

<u>Comparison with another model:</u>



```
Accuracy: 0.9262777404169469
F1 Score: 0.9164430952104438

Classification Report:
              precision    recall  f1-score   support

   Detractor       0.95      0.98      0.96      9790
     Neutral       0.41      0.04      0.08       295
    Promoter       0.81      0.81      0.81      1811

    accuracy                           0.93     11896
   macro avg       0.72      0.61      0.62     11896
weighted avg       0.91      0.93      0.92     11896
```

Figure 30 - Logistic Regression



```
Accuracy: 0.92
F1 Score: 0.9136166470459683
Classification Report:
              precision    recall  f1-score   support

   Detractor       0.96      0.96      0.96      9790
     Neutral       0.16      0.12      0.14       295
    Promoter       0.78      0.83      0.80      1811

    accuracy                           0.92     11896
   macro avg       0.63      0.64      0.63     11896
weighted avg       0.91      0.92      0.91     11896
```

Figure 31 – Support Vector Machine

Based solely on the provided metrics, it appears that the logistic regression model outperforms the SVM model across various evaluation criteria. Here's a breakdown based on precision, recall, and F1-score:

Precision:

- For the detractor class, logistic regression achieves a precision of 0.95, which is slightly higher than the precision of 0.96 achieved by the SVM model.
- In the promoter class, logistic regression achieves a precision of 0.81, outperforming the SVM model's precision of 0.78.
- Although the precision for the neutral class is higher for SVM (0.16) compared to logistic regression (0.41), the neutral class may have fewer instances or may be more challenging to predict accurately.

Recall:

- Logistic regression exhibits a higher recall across all classes compared to the SVM model. For example, in the neutral class, logistic regression achieves a recall of 0.04, whereas SVM only achieves a recall of 0.12. This indicates that logistic regression captures more instances of each class, which is particularly important for a balanced prediction.

F1-score:

- The F1-score, which balances precision and recall, reflects the overall performance of the model. Logistic regression consistently achieves higher F1-scores across all classes compared to the SVM model. For instance, while the F1-score for the neutral class is 0.08 for logistic regression and 0.14 for SVM, logistic regression generally achieves better F1-scores across all classes, including the detractor and promoter classes.

In summary, based on the provided metrics, logistic regression demonstrates superior performance over the SVM model in terms of precision, recall, and F1-score across all classes. It achieves higher precision and recall rates, indicating better accuracy in classifying instances. Additionally, the F1-score, which considers both precision and recall, also favours logistic regression, suggesting a better balance between precision and recall. Therefore, based solely on these metrics, logistic regression appears to be the better model for this particular classification task.

Discussion

Challenges faced as a group:

- **Managing Tight Deadlines:** Closer to Chinese New Year, it was more difficult to find a time to meet amongst the team to review each other's work. This is because everyone had different schedules to attend to, solved by finding the tiny pockets of time we all have and picked the ones that are most convenient for the entire group, implemented efficient task prioritization and clear delegation to overcome time constraints and avoid any impediments to our workflow.
- **Streamlining Team Meetings:** Initially, regular meetings proved to be time-consuming without substantial productivity gains. Optimized our approach by scheduling meetings only when necessary to discuss progress and critical issues which resulted in a more focused and efficient approach.

Challenges faced as individuals:

KARTHIK	JUN MING	PIN SHIEN
• Frequent kernel crashes required repetitive model retraining, leading to time wastage. Implemented serialization of models using pickle to save and reload trained models, preventing loss of progress and reducing downtime.	• Models like SVM take a very long time to train. Tuning also took a long time. Solved by letting the notebook access all my RAM and CPU that is not in used to speed up the code.	• Choosing the appropriate models for this project was difficult as I was unsure what would be a good fit for the data. Solved through proper research of the models and cutting down on irrelevant models
• Increasing code complexity and size made it difficult to manage and scale with new feature or model additions. Added markdown blocks for better code organization and stored all the models in a list, iterating over them for training and result	• Had issues generating ROC curve due to it requiring binary values. Solved by approaching teammates and online resource.	• Determining the appropriate pre-processing techniques to apply to my dataset was also difficult as I had not learnt some of the techniques that would be suitable. Solved by reading up online resources to

compilation, improving maintainability.		implement SMOTE as we didn't really touch on it
• Lengthy text cleaning processes lacked progress indication, making it difficult to estimate completion time or verify proper execution. Integrated a progress bar on the text cleaning process allowing for better process management.	• Messy code, for example, in the preprocessing of the test data, instead of putting it into the Text Classification notebook, placed and submitted in another notebook so that the flow seems more linear and clearer.	• Understanding the right evaluation metrics to use was also difficult as there were many metrics to choose from. Solved by discussing it with the team members

Conclusion

Based on the information provided for each model's performance on test data, the best model for predicting the detractor class with a high degree of accuracy is Karthik's SVM model. This model achieves perfect recall (100%) and precision (100%) for the detractor class, indicating that it can identify all detractors without any false positives, which is exceptional and fully meets the business goal of accurately pinpointing dissatisfied customers.

Additionally, Karthik's SVM model also performs excellently for the promoters and passives classes, with high scores across precision, recall, and F1-score. This indicates a strong overall performance and suggests that the model is well-rounded and effective across different customer sentiments.

In terms of business goals, which typically prioritize the correct identification of detractors to mitigate negative impacts and enhance customer satisfaction, Karthik's model not only achieves this but also excels in correctly identifying promoters and passives. This comprehensive and consistent performance across all classes suggests that the business goal is indeed achieved with this model.