SPRINGBOARD INTERNSHIP 4.0

Internship Report

“Emotions Classification Based on Feedback for an Event Planning Company using Text Classification”

Submitted by:
YASHAS M
(Project Intern)

Under the Guidance of:
SUDHEER KUMAR
(Mentor)
INFOSYS

Table of Contents

SL.NO Page No

CHAPTER 1 INTRODUCTION

In the dynamic and client-centric industry of event planning, understanding and responding to client emotions is crucial for delivering exceptional services and ensuring client satisfaction. Feedback from clients provides valuable insights into their experiences and emotional responses to events. However, the traditional methods of manually analysing this feedback are time-consuming, labour-intensive, and often subjective. These limitations can lead to inefficiencies and misinterpretations, hindering an event planning company's ability to swiftly address client concerns and enhance their services.

To overcome these challenges, the integration of machine learning techniques, particularly text classification, offers a promising solution. Text classification involves categorizing textual data into predefined labels—in this case, various emotions such as happiness, frustration, surprise, and disappointment. By employing machine learning algorithms, it is possible to automate the analysis of client feedback, making the process more efficient, objective, and scalable. This report explores the implementation of a machine learning-based text classification system to classify emotions in client feedback, aiming to provide event planners with actionable insights to improve service quality, enhance client experiences, and maintain a competitive edge in the industry.

CHAPTER 2 PROBLEM STATEMENT

In the event planning industry, client satisfaction and emotional engagement are crucial for success and long-term client relationships. Event planning companies receive a wealth of client feedback through various channels such as surveys, emails, social media, and online reviews. However, manually analysing this feedback to gauge client emotions is a labour-intensive and subjective process that is not scalable. This traditional approach can lead to inefficiencies and potential misinterpretations, hindering the company's ability to promptly address client concerns and improve service quality.

The objective of this project is to develop a machine learning-based text classification system to automate the classification of emotions in client feedback. The system must accurately identify a range of emotions, including sadness, joy, love, anger, fear and surprise from textual data. By leveraging machine learning algorithms such as Support vector machine, logistic regression, Random Forest and Xgboost , this project aims to provide a scalable, efficient, and objective solution for emotion analysis. This will enable the event planning company to gain deeper insights into client experiences, improve service quality by addressing negative feedback proactively, and enhance overall client satisfaction and retention.

CHAPTER 3 DATASET OVERVIEW

Datasets consist of following column:

1.Label: Integer

2.Text: String

The total length of the dataset is 16001 data. It consists of labels such as 0: sadness, 1: joy, 2: love, 3: anger, 4: fear, 5: surprise. The count of sadness is: 4666, joy is: 5362, love is: 1304, anger is: 2159, fear is: 1937, surprise is: 572.
Some categories, like Surprise and Love, have fewer data compared to others, like Joy and Sadness, this imbalance needs to be addressed during model training. This is important to ensure the classifier works well for all emotions. To handle this imbalance, techniques such as adding more data, adjusting the number of samples.
Initially we have worked on the **sentimental analysis** using different dataset which provide us the better understanding about the text classification.

Data Preprocessing Techniques:

- **Lower Case**: Convert all text to lowercase to ensure consistency and avoid duplication of words that differ only in case.

- **Remove Links**: Eliminate any hyperlinks present in the text as they typically do not contribute to the meaning of the text and can be distracting.

- **Remove New Lines (\n)**: Strip out newline characters to ensure that text is continuous and easier to process.

- **Words Containing Numbers**: Exclude words that contain numbers since they often represent identifiers or measurements rather than meaningful words.

- **Extra Spaces**: Trim excess spaces to standardize the text and ensure consistent tokenization.

- **Special Characters**: Remove special characters like punctuation, symbols, or emojis, as they may not contribute to the semantics of the text.

• **Removal of Stop Words**: Filter out common words (stop words) like "the," "is," "and" which occur frequently but often do not carry significant meaning.

• **Stemming**: Reduce words to their base or root form, removing suffixes and prefixes. For example, "running" becomes "run." This helps in reducing the dimensionality of the feature space and capturing the essence of the word. Popular stemming algorithms include Porter Stemmer, Snowball Stemmer, and Lancaster Stemmer.

• **Lemmatization**: Similar to stemming, but instead of just removing prefixes and suffixes, lemmatization maps words to their base or dictionary form (lemma). For example, "better" becomes "good." This approach ensures that the transformed words are valid lemmas, which can improve interpretability.

Chapter 4 Modelling Approach

1. Data (Text) Preprocessing:

• Lower Case

• Remove links

• Remove next lines (\n)

• Words containing numbers

• Extra spaces

• Special characters

• Removal of stop words

• Stemming

• Lemmatization

2. Featuring Engineering:

• Convert the Text corpus to a matrix of word counts. (Vectorize the Text data)
Example: Use Tf-IDF

3. Model Building:

1. Divide the dataset into Train (70%), Test (20%) and Validation (10%) datasets.
2. Build at least 3 classification models

Step 1: Build model 1 and generate the classification report (Performance metrics using Confusion Metrics) for both Training and Test datasets.

Step 2:

• Use grid search or binary search for Hyperparameter Tuning.

• Use at least 2 values for each hyperparameters.

• Choose the best model parameters based on grid search and generate the classification report (Performance metrics using Confusion Metrics) for both Training and Test datasets.

Step 3: Repeat step 1 and 2 for Model 2 and Model 3 as well.

Step 4: Now choose the final model based on the classification report (Performance metrics using Confusion Metrics) for both Training, Test and validation datasets.

4. Data Visualization: Input and Output plots

CHAPTER 5 Models used and Hyperparameter Tuning

I have implemented various models such as Random Forest, Support Vector Machine (SVM), Logistic Regression, and XGBoost algorithms along with hyperparameter tuning for each model using Grid Search to optimize model performance.

1. Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) of the individual trees. It is robust to overfitting and can handle large datasets with higher dimensionality.
Key Hyperparameters I have used:

- **n_estimators:** Number of trees in the forest.
- **max_depth:** Maximum depth of each tree.

FEATURES:

- **Feature Importance**: Provides estimates of feature importance, which can be very insightful for understanding the dataset.
- **Handles Missing Values**: Can handle missing data better than many other models like SVM and Logistic Regression.
- **Robustness**: More robust to noise and outliers compared to algorithms like k-nearest neighbors (KNN).

2. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful classifier that works by finding the hyperplane that best separates the classes in the feature space. SVMs are effective in high-dimensional spaces and are versatile with different kernel functions.

Key Hyperparameters I have used:

- **C:** Regularization parameter, balancing margin maximization and error term.
- **kernel:** Specifies the kernel type (e.g., linear, polynomial, radial basis function).

FEATURES:

- **Effective in High Dimensions**: Particularly effective when the number of dimensions exceeds the number of samples.
- **Versatility with Kernels**: Different kernel functions (linear, polynomial, RBF) allow SVM to model complex relationships.
- **Margin Maximization**: Focuses on maximizing the margin between classes, which enhances generalization.

3. Logistic Regression

Logistic Regression is a linear model for binary classification that estimates the probability of a binary response based on one or more predictor variables. It uses the logistic function to model a binary outcome.

Key Hyperparameters I have used:

- **C:** Inverse of regularization strength; smaller values specify stronger regularization.
- **solver:** Algorithm to use for optimization (e.g., liblinear, lbfgs)
- **max_iter:** Maximum number of iterations for the solvers to converge.

FEATURES:

- **Simplicity and Interpretability**: Easy to implement and interpret, making it suitable for understanding the relationship between features and the outcome.
- **Probabilistic Output:** Provides probability estimates, which are useful for binary classification tasks.
- **Less Prone to Overfitting:** With regularization (L1, L2), it can prevent overfitting better than complex models.

4. XGBoost

XGBoost (Extreme Gradient Boosting) is an efficient and scalable implementation of gradient boosting framework. It is designed for speed and performance and provides a robust model through boosting trees.

Key Hyperparameters I have used:

- **n_estimators:** Number of boosting rounds.
- **learning_rate:** Step size shrinkage used in update to prevent overfitting.

FEATURES:

- **Boosting Technique**: Uses boosting, which sequentially builds models to correct errors of previous ones, leading to high accuracy.
- **Regularization**: Includes built-in L1 and L2 regularization to prevent overfitting, which is not present in other boosting algorithms like AdaBoost.
- **Speed and Performance**: Highly optimized for speed and performance, often outperforming other algorithms in competitive machine learning tasks.
- **Handling Missing Data**: Automatically handles missing values effectively.

Hyperparameter Tuning Using Grid Search

Grid Search is a method for systematically working through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance. This approach helps in identifying the best combination of hyperparameters for each model.

Steps in Grid Search:

1. **Define the parameter grid**: Specify the range of hyperparameters to be tested.
2. **Cross-validation**: Split the training data into k folds and perform k-fold cross-validation for each combination of hyperparameters.
3. **Model fitting**: Train the model using each combination of hyperparameters on the training data.
4. **Performance evaluation**: Evaluate the model performance on the validation set and select the best-performing hyperparameters.
5. **Model selection**: Choose the model with the optimal set of hyperparameters based on cross-validation results.

CHAPTER 6 RESULTS

Performance Metrics

- **Accuracy**: The proportion of correctly classified instances out of the total instances in the dataset.
- **Precision**: The proportion of true positive predictions among all positive predictions made by the model.
- **Recall**: The proportion of true positive predictions among all actual positive instances in the dataset.
- **F1-score**: The harmonic mean of precision and recall, balancing the two metrics, especially important for imbalanced datasets.

Confusion Matrix

A confusion matrix is a tabular summary of the performance of a classification model, comparing the model's predictions to the actual ground truth labels. It provides detailed insights into the model's behavior and performance across different classes.

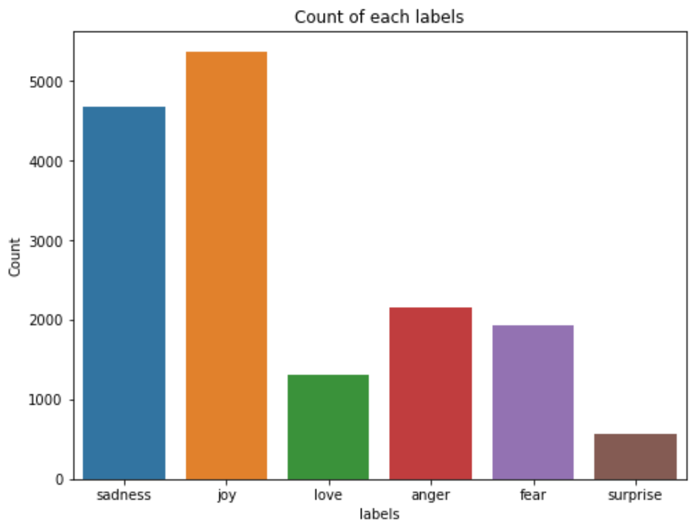


Fig 6.1. label wise comparison of dataset using bar graph

It consists of labels such as 0: sadness, 1: joy, 2: love, 3: anger, 4: fear, 5: surprise. The count of sadness is: 4666, joy is: 5362, love is: 1304, anger is: 2159, fear is: 1937, surprise is: 572.

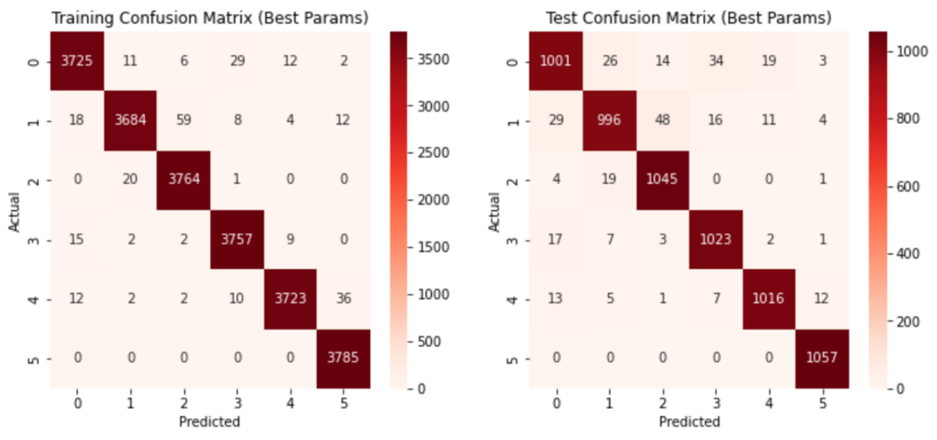


Fig 6.2. Confusion matrix of the logistic regression after hyperparameter tuning.

Training dataset: The highest number of correct classification is given for surprise label which is 3785 .highest number of wrong classification for the same class as in training dataset that is for label love and joy which is 59, the model predicted that it is love label but actual value is joy, this is decreased when compared to the model without hyperparameter tuning

Testing dataset: This gives highest number of wrong classifications for the same class as in training dataset that is for label love and joy which is 48, the model predicted that it is love label, but actual value is joy. The highest number of correct classifications is given for sur prise label which is 1057

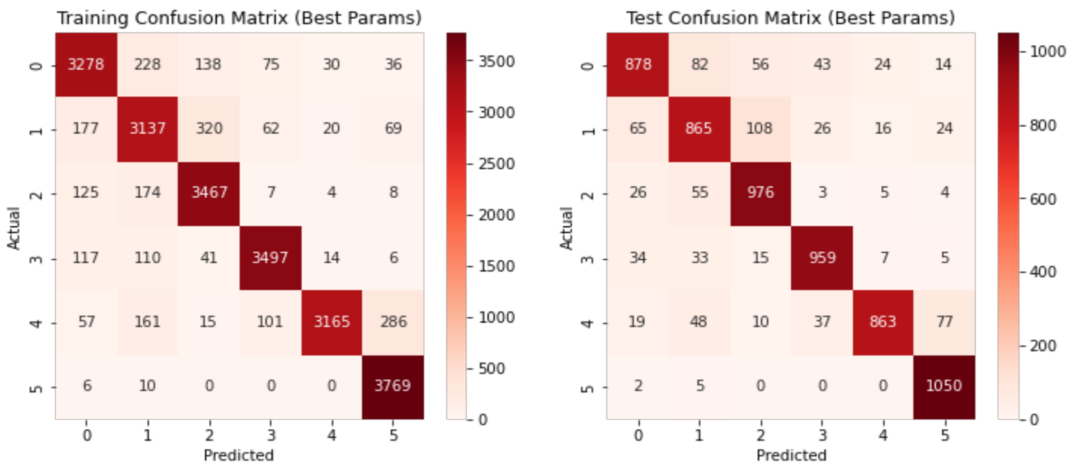


Fig 6.3. The confusion matrix of the random forest algorithm after the hyperparameter tuning

Training dataset: this gives highest number of wrong classifications for the same class as in training dataset that is for label fear and surprise which is 286, the model predicted that it is surprise label, but actual value is fear.
The highest number of correct classifications is given for surprise label which is 3769.

Testing dataset: This gives highest number of wrong classifications for the same class as in training dataset that is for label joy and love which is 108, the model predicted that it is love label, but actual value is joy.

The highest number of correct classifications is given for surprise label which is 1050.

This model generates a greater number of wrong classifications

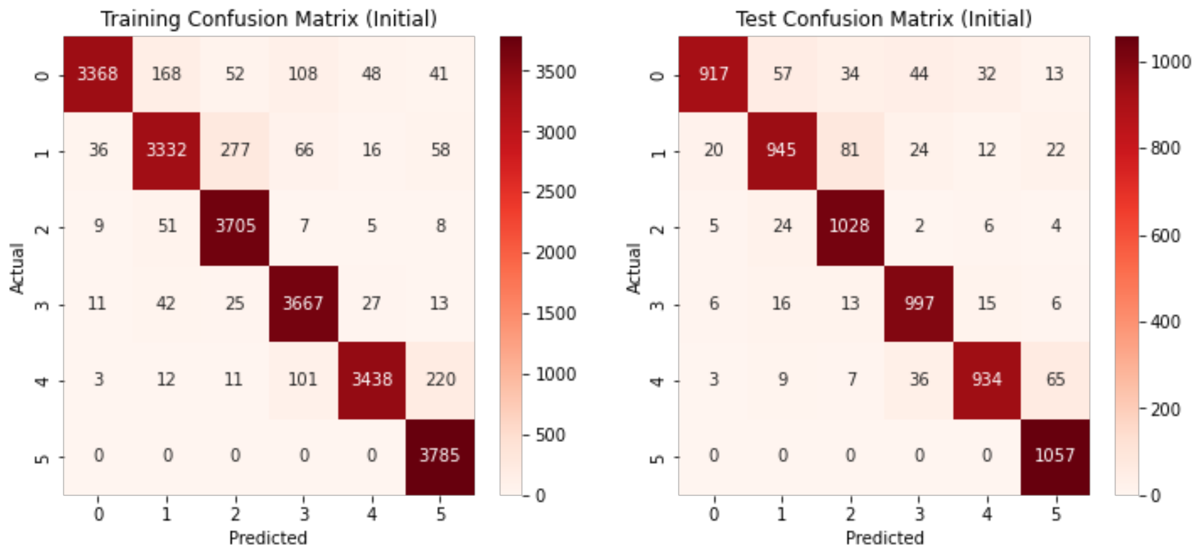
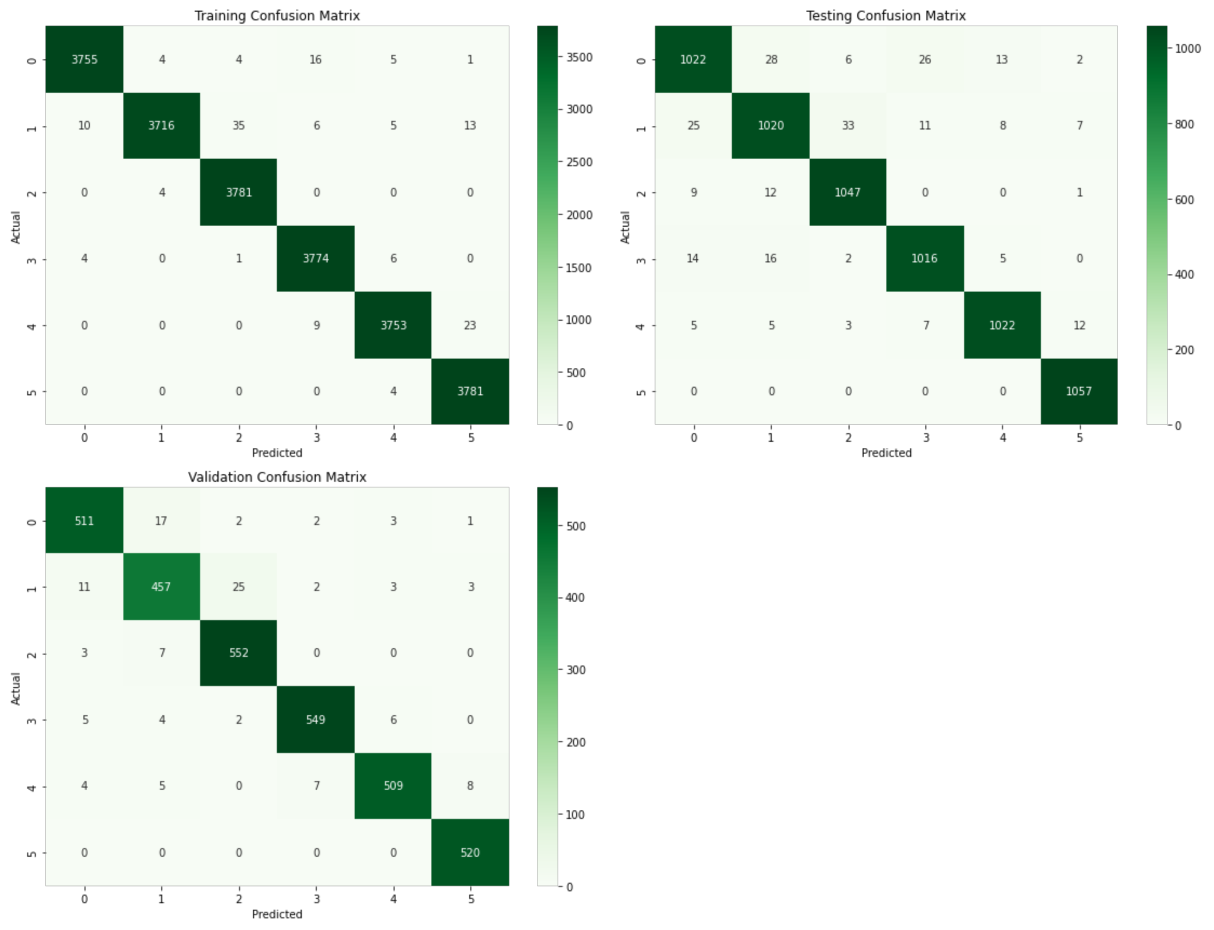


Fig 6.4. Confusion Matrix of the Xgboost algorithm after hyperparameter tuning.

Training dataset: the highest number of correct classifications is given for surprise label which is 3785. highest number of wrong classifications for the same class as in training dataset that is for label love and joy which is 277, the model predicted that it is love label, but actual value is joy.

Testing dataset: this gives highest number of wrong classifications for the same class as in training dataset that is for label love and joy which is 81, the model predicted that it is love label, but actual value is joy. the highest number of correct classifications is given for surprise label which is 1057

The amount of prediction which is wrongly classified is increased from normal model to hyperparameters tuning.



Training Confusion Matrix

Actual \ Predicted	0	1	2	3	4	5
0	3755	4	4	16	5	1
1	10	3716	35	6	5	13
2	0	4	3781	0	0	0
3	4	0	1	3774	6	0
4	0	0	0	9	3753	23
5	0	0	0	0	4	3781

Testing Confusion Matrix

Actual \ Predicted	0	1	2	3	4	5
0	1022	28	6	26	13	2
1	25	1020	33	11	8	7
2	9	12	1047	0	0	1
3	14	16	2	1016	5	0
4	5	5	3	7	1022	12
5	0	0	0	0	0	1057

Validation Confusion Matrix

Actual \ Predicted	0	1	2	3	4	5
0	511	17	2	2	3	1
1	11	457	25	2	3	3
2	3	7	552	0	0	0
3	5	4	2	549	6	0
4	4	5	0	7	509	8
5	0	0	0	0	0	520

Fig 6.5. Confusion matrix of the SVM algorithm after hyperparameter tuning

Training dataset: the highest number of correct classifications is given for surprise, Joy label which is 3781 highest number of wrong classifications for the same class as in training dataset that is for label love and joy which is 35, the model predicted that it is love label, but actual value is joy.

Testing dataset: This gives highest number of wrong classifications for the same class as in training dataset that is for label love and joy which is 33, the model predicted that it is love label, but actual value is joy.

the highest number of correct classifications is given for surprise label which is 1057.

Validation dataset: this gives highest number of wrong classifications for the same class as in training dataset that is for label love and joy which is 25, the model predicted that it is love label, but actual value is joy.

The highest number of correct classifications is given for love label which is 552.

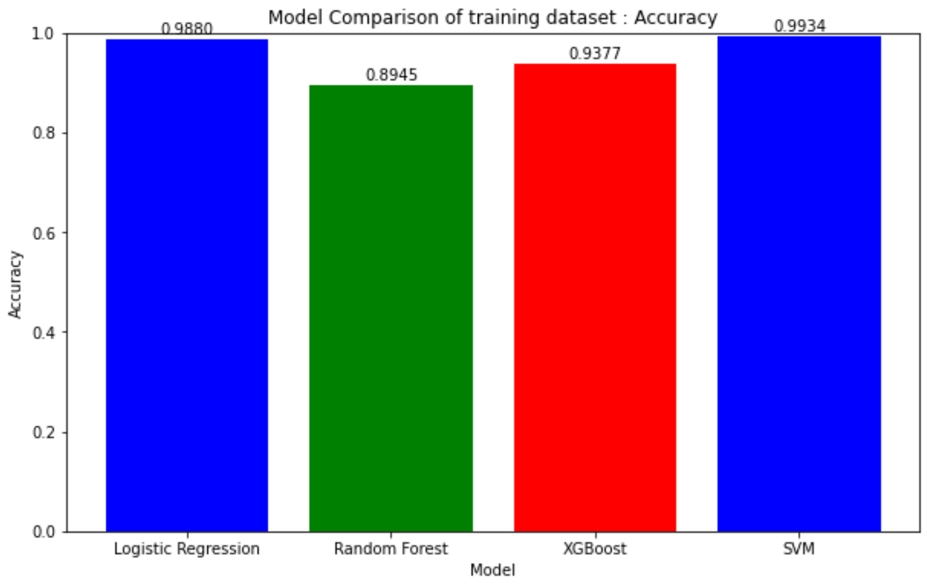


Fig 6.6. Model wise comparison based on Accuracy for training dataset

Model	Performance Score
Logistic Regression	0.988
Random Forest	0.894
XGBoost	0.938
SVM	0.993

The SVM model provides the highest accuracy that is 99% (approx.), whereas logistic regression, random forest and XGboost algorithm generates accuracy 98%,89% and 93% respectively.

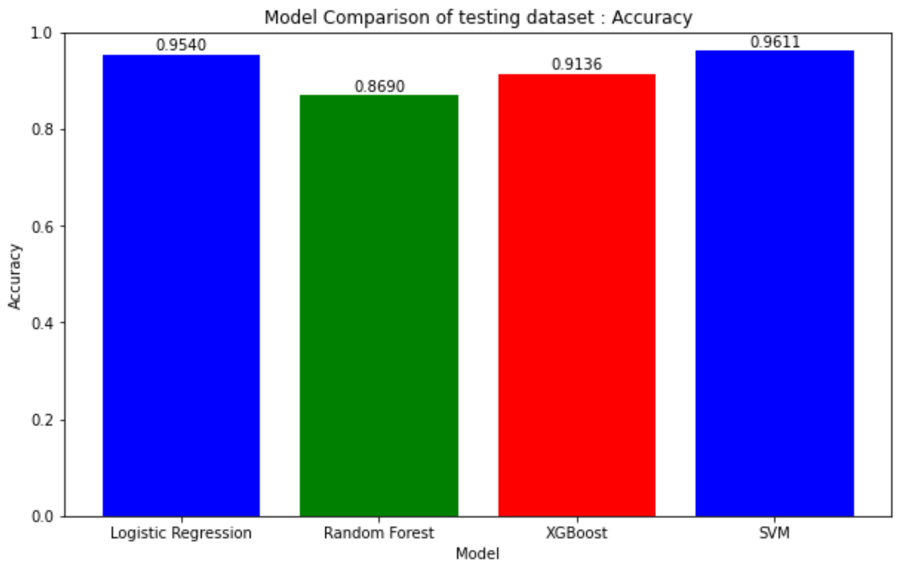
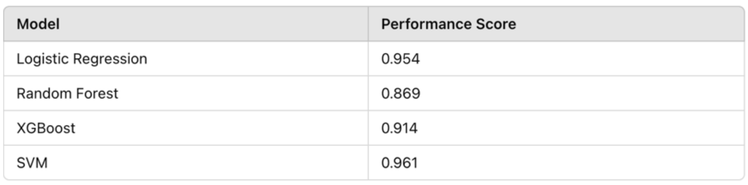


Fig 6.7. Model wise comparison based on Accuracy for testing dataset.



Model	Performance Score
Logistic Regression	0.954
Random Forest	0.869
XGBoost	0.914
SVM	0.961

The SVM model provides the highest accuracy that is 96%(approx.), whereas logistic regression, random forest and Xgboost algorithm generates accuracy 95%, 86% and 91% respectively.

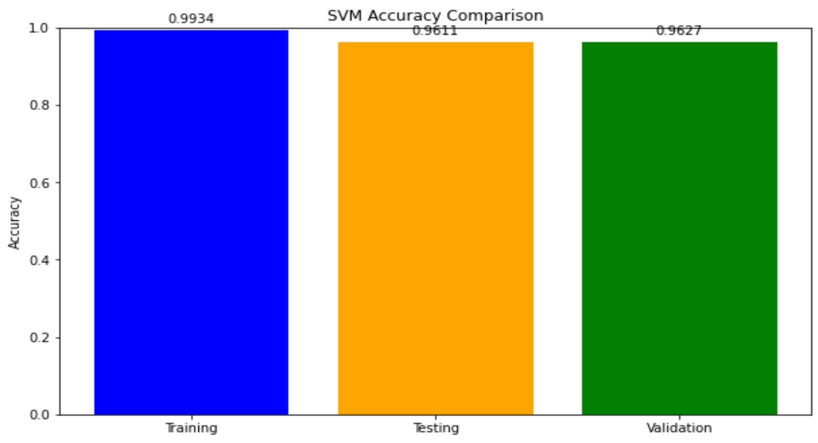


Fig 6.8.SVM model comparison between Training, Testing and Validation dataset.

SVM is best model which provides the average accuracy of 97% where the training dataset accuracy is 99.3%, testing dataset is 96% accuracy and Validation dataset provides 96%

CHAPTER 7 RECOMMENDATION AND RESOLUTION

Based on the evaluation of our machine learning models, we recommend implementing the Support Vector Machine (SVM) as the primary model for emotion classification in client feedback due to its highest test accuracy of 96% and average accuracy of 97.73%. SVM's robust generalization capability ensures reliable emotion detection across diverse feedback datasets. Additionally, Logistic Regression, with its strong performance and interpretability, should be used to provide actionable insights into the relationship between input features and emotions. XGBoost, offering balanced performance and effective handling of complex data, can complement SVM, particularly for negative feedback. Although Random Forest had the lowest accuracy, its ability to identify significant features makes it a valuable addition to the emotion classification system.

To operationalize these models, we recommend developing an integrated system architecture that automates feedback analysis, making it scalable and efficient. This system should include a standardized data pipeline for collecting and preprocessing feedback from various channels, ensuring high data quality. An intuitive dashboard should be created for visualizing emotion analysis results and tracking trends, along with real-time alert systems for immediate intervention using tools such as PowerBi, tableau additionally. By leveraging these models, the company can promptly address negative feedback, improve service quality, and enhance overall client satisfaction and retention. Regular performance monitoring and user feedback collection will be essential for ongoing system improvement and effectiveness.

CONCLUSION

In this project we have used four models such as Logistic Regression, Random Forest, XGBoost, and Support Vector Machine (SVM). The evaluation was based on test accuracy, training accuracy, and average accuracy. Among these models, SVM stood out as the best performer, achieving the highest test accuracy of 96% and an impressive average accuracy of 0.9773. This indicates SVM's strong generalization capability and robustness in handling the classification task.

Logistic Regression also demonstrated solid performance with a test accuracy of 95% and an average accuracy of 97%, making it a reliable choice for tasks where model interpretability and linear relationships are important. XGBoost, with a test accuracy of 91% and an average accuracy of 92%, showed balanced performance, benefiting from its boosting technique to enhance accuracy and manage overfitting. Random Forest, although the lowest in test accuracy 86.9% and average accuracy 88.17%, still provided valuable insights into feature importance and demonstrated robustness. Based on these results, SVM is identified as the best performing model, offering the highest accuracy and generalization among the evaluated models.

This model help event planning company to rework on their services and make sure they meet the expectations of the clients in order to get good number of contracts in future as well as to retain the clients. additionally. By leveraging these models, the company can promptly address negative feedback, improve service quality, and enhance overall client satisfaction and retention. Regular performance monitoring and user feedback collection will be essential for ongoing system improvement and effectiveness.