|  |  |
| --- | --- |
| The data will set you free. What does Twitter sentiment analysis say about  major Airlines? - The World As Perpetual Beta  **SENTIMENT ANALYSIS OF AIRLINE PASSENGERS’ TWEETS**  **GROUP 5**  Ankit Tripathi  Nitin Shrimal  Dinesh Prakash Singh | ABSTRACT  The key aim of this project is to perform text & sentiment analysis on extracted airline travel tweets that resulted in a positive or negative. |

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**Work Integrated Learning Programmes Division**

**Post Graduate Program**

**in**

**Artificial Intelligence and Machine Learning**

SENTIMENT ANALYSIS OF AIRLINE PASSENGERS’ TWEETS

CAPSTONE PROJECT

Submitted in partial fulfilment of the requirements of the

Post Graduate Certification Program

in

Artificial Intelligence and Machine Learning

By

ANKIT TRIPATHI (2022AIML554)

NITIN SHRIMAL (2022AIML528)

DINESH PRAKASH SINGH (2022AIML510)

Under the supervision of

Gautham Gangopadhyay

Project work carried out at

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

Pilani (Rajasthan) INDIA

(Feb-Mar, 2024)

# **Acknowledgements**

We would like to express our deepest gratitude to Mr. Gautham Gangopadhyay for his invaluable guidance and support throughout the duration of this project. Mr. Gangopadhyay played a pivotal role in helping us comprehend the intricacies of sentiment analysis of airline passengers' tweets. His expertise enabled us to understand the project scope, the underlying data, and the desired end goal with clarity. We are especially grateful for his generous assistance in conducting code reviews, providing insightful feedback, and sharing sample code for reference, which significantly contributed to the development of our analysis.

We would also like to extend our sincere appreciation to Swarna mam for her constructive feedback during the mid-semester review. Her inputs were instrumental in refining our model performance and enhancing the quality of our final outcomes.

Additionally, we would like to thank Mr. Parthasarathy, our TA, for his diligent efforts in coordinating and facilitating all meetings, ensuring smooth progress and effective communication throughout the project timeline.

Moreover, we express our gratitude to BITS Pilani for providing us with the opportunity to delve into the realms of AI/ML, fostering an environment conducive to learning and innovation.

We are truly grateful to each of these individuals and institutions for their unwavering support, guidance, and encouragement, which have been invaluable assets in the successful completion of this project.

# **Certificate from the Mentor**

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**CERTIFICATE**

This is to certify that the Capstone Project entitled\_\_ SENTIMENT ANALYSIS OF AIRLINE PASSENGERS’ TWEETS

and submitted by Mr. ANKIT TRIPATHI, Mr. NITIN SHRIMAL, Mr. DINESH PRAKASH SINGH ID Nos. 2022AIML554, 2022AIML528, 2022AIML510 respectively

in partial fulfilment of the requirements of PCAM ZC321 Capstone Project, embodies the work

done by him/her under my supervision.

A picture containing clock

Description automatically generated

Place: Kolkata Signature of the Mentor

Date: 15th Mar 2024 Name : GAUTHAM GANGOPADHYAY

# **Dissertation Abstract**

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**PROJECT REPORT 2023-24**

**PCAM ZC321 CAPSTONE PROJECT**

Project Title : \_\_ SENTIMENT ANALYSIS OF AIRLINE PASSENGERS’ TWEETS

Name of Mentor : \_\_GAUTHAM GANGOPADHYAY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Student(s) : \_ANKIT TRIPATHI, NITIN SHRIMAL, DINESH PRAKASH SINGH

ID No. of Student(s) : \_ 2022AIML554, 2022AIML528, 2022AIML510

## Abstract

In today's digitally driven landscape, the ability to derive actionable insights from vast amounts of data is crucial for businesses to stay competitive. This project presents a comprehensive end-to-end solution for sentiment analysis of airline passengers' tweets, leveraging advanced technologies such as Streamlit and Apache Airflow to deliver seamless and efficient functionality.

The Streamlit application serves as a user-friendly interface, allowing customers to interact with the sentiment analysis tool effortlessly. Through this application, users can input tweets or select predefined datasets, and instantly receive insightful analysis regarding the sentiments expressed by airline passengers. The intuitive design and functionality of the Streamlit application ensure a seamless user experience, enabling customers to derive valuable insights with ease.

Additionally, the model creation process involves meticulous steps to ensure robustness and accuracy. Initially, the data is collected and preprocessed to prepare it for analysis. Various machine learning algorithms, including Logistic Regression, Decision Trees, Random Forest, and others, are evaluated to determine the most suitable model for sentiment analysis. Hyperparameter tuning is performed to optimize model performance further. Once the best model is selected, it is trained on the training dataset and evaluated using the test dataset to assess its generalization ability accurately.

Furthermore, the orchestration of the sentiment analysis pipeline using Apache Airflow ensures the model remains up-to-date and relevant. By scheduling daily data ingestion from customers and refreshing the model accordingly, Apache Airflow automates the process of keeping the sentiment analysis framework current. This orchestration not only enhances the accuracy and reliability of the analysis but also ensures that customers always have access to the latest insights.

Through the integration of Streamlit and Apache Airflow, this project offers a comprehensive solution for sentiment analysis of airline passengers' tweets, empowering businesses to make data-driven decisions and enhance customer satisfaction effectively. This end-to-end approach demonstrates the transformative potential of advanced technologies in driving innovation and excellence in the realm of data analytics.

# **List of Symbols & Abbreviations Used**

# **List of Tables**

* LogisticRegression(): Logistic Regression Classifier
* DecisionTreeClassifier(): Decision Tree Classifier
* RandomForestClassifier(): Random Forest Classifier
* GaussianNB(): Gaussian Naive Bayes Classifier
* AdaBoostClassifier(): AdaBoost Classifier
* GradientBoostingClassifier(): Gradient Boosting Classifier
* KNeighborsClassifier(): K-Nearest Neighbors Classifier
* XGBClassifier(): Extreme Gradient Boosting Classifier
* LinearSVC(): Linear Support Vector Classifier
* Deep Learning using Bart: Bart-based Deep Learning Model

# **Table of Contents**

[**Acknowledgements** 3](#_Toc161308006)

[**Certificate from the Mentor** 4](#_Toc161308007)

[**Dissertation Abstract** 5](#_Toc161308008)

[Abstract 6](#_Toc161308009)

[**List of Symbols & Abbreviations Used** 7](#_Toc161308010)

[**List of Tables** 7](#_Toc161308011)

[**Table of Contents** 8](#_Toc161308012)

[**Project Details** 9](#_Toc161308013)

[1.1. Problem statement 9](#_Toc161308014)

[1.2. Objective 10](#_Toc161308015)

[1.3. Business Benefits 10](#_Toc161308016)

[**Project Specifications** 12](#_Toc161308017)

[1.4. Machine Learning Process Flow 12](#_Toc161308018)

[1.5. Project Resources 14](#_Toc161308019)

[1.6. Potential Data Challenges & Risks 14](#_Toc161308020)

[1.7. Detailed Project Plan 15](#_Toc161308021)

[1.8. Approach 17](#_Toc161308022)

[**Model Development** 18](#_Toc161308023)

[1.9. Feature Engineering / Data Pre-processing activities 18](#_Toc161308024)

[1.10. Generating Machine Learning Data models 18](#_Toc161308025)

[1.11. Libraries 18](#_Toc161308026)

[1.12. Test/Train Split 19](#_Toc161308027)

[1.13. SKLearn Models 20](#_Toc161308028)

[1.14. SKLearn models for multiclass classification 20](#_Toc161308029)

[1.15. SKLearn models for Binary classification 28](#_Toc161308030)

[1.16. SKLearn models for Binary classification – Hyper Parameter Tuning 29](#_Toc161308031)

[**Conclusions / Recommendations** 30](#_Toc161308032)

[**Future Work & Extension or Scope of improvements** 31](#_Toc161308033)

[**Bibliography / References** 32](#_Toc161308034)

[**Appendices** 32](#_Toc161308035)

# **Project Details**

## Problem statement

The aviation industry is highly reliant on customer satisfaction and feedback to maintain competitiveness and improve service quality. With the proliferation of social media platforms, airline passengers often express their opinions, experiences, and sentiments through tweets, providing a valuable source of feedback for airlines. However, manually analysing large volumes of tweets to gauge customer sentiment is a labour-intensive and time-consuming process.

The problem statement for this project is to develop an automated sentiment analysis solution for airline passengers' tweets. The objective is to leverage machine learning algorithms and natural language processing techniques to accurately classify tweets into positive, negative, or neutral sentiments. By automating the sentiment analysis process, airlines can efficiently monitor customer feedback in real-time, identify emerging trends, and promptly address issues to enhance customer satisfaction.

The key challenges to address in this project include:

* **Data Collection and Preprocessing**: Gathering a comprehensive dataset of airline passengers' tweets and preprocessing the data to remove noise, handle missing values, and standardize text format.
* **Model Selection and Training**: Exploring various machine learning algorithms and deep learning techniques to identify the most suitable approach for sentiment analysis. This involves training and evaluating multiple models to achieve optimal performance.
* **Evaluation Metrics**: Establishing appropriate evaluation metrics to assess the accuracy, precision, recall, and F1-score of the sentiment analysis models.
* **Deployment and Integration**: Developing a user-friendly interface, such as a web application, to allow stakeholders to interact with the sentiment analysis tool. Additionally, integrating the sentiment analysis solution with existing systems or workflows within the airline industry.

By addressing these challenges, the project aims to provide airlines with a robust and scalable solution for sentiment analysis of passengers' tweets, enabling them to proactively respond to customer feedback and improve overall service quality.

## Objective

The primary objective of this project is to develop an automated sentiment analysis solution for airline passengers' tweets. This involves leveraging machine learning algorithms and natural language processing techniques to accurately classify tweets into positive, negative, or neutral sentiments. The key objectives include:

* **Develop a Comprehensive Dataset**: Gather a comprehensive dataset of airline passengers' tweets from various sources, including social media platforms and customer feedback channels.
* **Preprocess the Data**: Preprocess the collected data to remove noise, handle missing values, standardize text format, and perform other necessary data cleaning tasks.
* **Explore and Select Models**: Explore a range of machine learning algorithms and deep learning techniques for sentiment analysis, including Logistic Regression, Decision Trees, Random Forests, Naive Bayes, Boosting algorithms, K-Nearest Neighbors, Extreme Gradient Boosting, Support Vector Machines, and Deep Learning models.
* **Train and Evaluate Models**: Train multiple sentiment analysis models using the selected algorithms and evaluate their performance using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.
* **Optimize Model Performance**: Fine-tune the selected models and optimize their hyperparameters to achieve optimal performance in classifying tweets into positive, negative, or neutral sentiments.
* **Develop a User-Friendly Interface**: Develop a user-friendly interface, such as a web application, to allow stakeholders (e.g., airline operators, customer service representatives) to interact with the sentiment analysis tool easily.
* **Deploy and Integrate the Solution**: Deploy the sentiment analysis solution in a production environment and integrate it with existing systems or workflows within the airline industry to enable real-time monitoring of customer feedback.
* **Provide Actionable Insights**: Provide actionable insights derived from sentiment analysis to airline operators, enabling them to identify emerging trends, address customer concerns promptly, and improve overall service quality.

By achieving these objectives, the project aims to empower airlines with an efficient and scalable solution for sentiment analysis of passengers' tweets, ultimately enhancing customer satisfaction and driving operational excellence in the aviation industry.

## Business Benefits

The implementation of an automated sentiment analysis solution for airline passengers' tweets offers numerous business benefits to airlines and other stakeholders in the aviation industry:

* **Enhanced Customer Satisfaction**: By promptly identifying and addressing customer concerns expressed through tweets, airlines can improve overall customer satisfaction levels. Proactive responses to feedback demonstrate attentiveness to customer needs and foster a positive brand image.
* **Real-Time Feedback Monitoring**: The automated sentiment analysis solution enables airlines to monitor customer feedback in real-time, allowing them to stay informed about emerging trends, issues, and sentiment shifts. This timely feedback facilitates agile decision-making and swift responses to customer concerns.
* **Improved Service Quality**: Insights derived from sentiment analysis help airlines identify areas for improvement in service delivery, onboard experience, and customer interactions. By addressing pain points and enhancing service quality, airlines can differentiate themselves in a competitive market and retain loyal customers.
* **Strategic Marketing and Communication**: Understanding customer sentiment allows airlines to tailor their marketing messages and communication strategies effectively. By aligning messaging with customer preferences and sentiment, airlines can resonate more deeply with their target audience and drive engagement.
* **Competitive Advantage**: Airlines that leverage sentiment analysis to understand and respond to customer feedback gain a competitive edge in the market. By demonstrating a commitment to customer satisfaction and continuous improvement, airlines can attract new customers and retain existing ones, thereby increasing market share.
* **Crisis Management and Risk Mitigation**: Sentiment analysis enables airlines to detect and mitigate potential crises or negative publicity early on. By monitoring sentiment trends and addressing issues proactively, airlines can prevent minor incidents from escalating into larger PR crises, protecting their reputation and brand integrity.
* **Data-Driven Decision Making**: Sentiment analysis provides airlines with valuable insights derived from customer feedback data. These insights empower decision-makers to make informed, data-driven decisions across various areas, including product development, marketing strategies, and customer service initiatives.
* **Cost Savings and Operational Efficiency**: Automating the sentiment analysis process reduces the time and resources required for manual analysis of tweets. By streamlining feedback monitoring and analysis, airlines can achieve cost savings and improve operational efficiency, allowing them to focus resources on other strategic initiatives.

Overall, the implementation of an automated sentiment analysis solution offers significant business benefits to airlines, including enhanced customer satisfaction, improved service quality, competitive advantage, and cost savings. By leveraging customer feedback effectively, airlines can drive positive outcomes and foster long-term success in the dynamic aviation industry.

# **Project Specifications**

## Machine Learning Process Flow

Ankit Tripathi, Nitin Shrimal, and Dinesh Prakash Singh conducted sentiment analysis based on the Six USA Airlines dataset. The research architecture comprised several stages, including data collection, data preprocessing, feature engineering, data splitting, lexicon-based sentiment analysis, application of deep learning and machine learning algorithms, data prediction, and model evaluation. The dataset encompassed three types of sentiments: positive, negative, and neutral. Leveraging the Vader model contributed to enhancing classification accuracy. However, it's worth noting that neutral sentiment is not taken into consideration in the evaluation metrics.

**Data Collection**: The process begins with the collection of relevant data from various sources, which could include structured datasets, unstructured text data, images, or other forms of data depending on the task at hand.

**Data Preprocessing**: Once the data is collected, it undergoes preprocessing to clean and transform it into a suitable format for analysis. This step may involve handling missing values, encoding categorical variables, scaling numerical features, and other data cleaning operations.

**Feature Engineering**: Feature engineering aims to create new features or transform existing ones to improve the performance of machine learning models. Techniques such as dimensionality reduction, feature selection, and creating interaction terms may be employed in this step.

**Data Splitting**: The dataset is divided into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune hyperparameters and evaluate model performance during training, and the test set is used to assess the final performance of the trained model.

**Model Selection**: Various machine learning algorithms are explored and evaluated to select the most suitable model for the task. This involves comparing different algorithms based on their performance metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC).

**Hyperparameter Tuning**: Once the model is selected, its hyperparameters are tuned to optimize its performance. Techniques such as grid search, random search, or Bayesian optimization may be used to find the best combination of hyperparameters.

**Model Training**: The selected model is trained using the training dataset. During training, the model learns the patterns and relationships present in the data, adjusting its parameters to minimize the error between predicted and actual values.

**Model Evaluation**: After training, the model is evaluated using the validation dataset to assess its performance. This step helps identify potential issues such as overfitting or underfitting and allows for further refinement of the model.

**Model Deployment**: Once the model is trained and evaluated, it is deployed into a production environment where it can be used to make predictions on new, unseen data. This could involve integrating the model into an application or system where it can generate predictions in real-time.

**Monitoring and Maintenance**: After deployment, the model is monitored to ensure its continued performance and accuracy. This may involve tracking key performance metrics, detecting drift in data distribution, and retraining the model periodically with new data to maintain its effectiveness over time

## Project Resources

**People:**

Following Resources worked on this project for duration of 8 weeks.

* Ankit Tripathi
* Nitin Shrimal
* Dinesh Prakash Singh

**Hardware**:

* 8GB RAM
* Windows 10
* Intel Core i5

**Software**:

* Jupyter Notebook
* Anaconda
* Python
* Nltk
* Vader
* Wordcloud
* Streamlit
* Apache Airflow
* Scikit-learn
* Matplotlib
* Pytorch

**Communication Channel**:

* Google Meet
* Google drive
* WhatsApp

## Potential Data Challenges & Risks

* **Missing Target Values**: The initial dataset provided lacked target values indicating sentiment labels, necessitating the use of sentiment analysis tools such as TextBlob and Vader to infer sentiment polarity. However, this approach may introduce inaccuracies in labelling due to the subjective nature of sentiment analysis and variations in tool performance.
* **Inherent Limitations of Sentiment Analysis Tools**: TextBlob and Vader, while widely used, may not always provide 100% accurate sentiment scores. These tools rely on predefined lexicons and heuristics to determine sentiment, which may not capture the nuances of context-specific language or sarcasm, leading to potential misclassifications.
* **Dependency on External Sentiment Analysis Tools**: Relying solely on external sentiment analysis tools introduces a dependency on third-party services, which may have associated costs or reliability issues. Additionally, updates or changes to these tools could impact the accuracy and consistency of sentiment labelling.
* **Lack of Client-Provided Target Values**: While sentiment analysis tools can infer sentiment labels, having client-provided target values would offer more relevance and accuracy in sentiment classification. Collaborating closely with clients to obtain labelled data could mitigate the risks associated with automated sentiment labelling.
* **Imbalanced Dataset**: Another challenge is the presence of an imbalanced dataset, where there may be fewer negative tweets compared to positive or neutral tweets. This imbalance can skew model predictions and lead to biased outcomes, requiring techniques such as oversampling, under-sampling, or the use of class weights to address.

## Detailed Project Plan

Following is the detailed project plan to successfully complete the project.

|  |  |  |
| --- | --- | --- |
| Week # | List of Activities | Status |
| Week# 1 & 2 | 1. Introduction to the project, team members & mentor. 2. Understanding the problem statement 3. Understanding the business benefits 4. Environment Setup 5. Planning the tasks 6. Discussion with team members 7. Establishing root folder to contain code, training tweets and creation of output files through scripting 8. Finding the missing values in each column 9. Drop rows which contain NULL tweets as Text Processing will be done on the tweet column 10. Drafting PPT document and review 11. Discussion with team member(Code Review) 12. Review with the mentor | Completed |
| Week# 3 & 4 | 1. Identifying and removing duplicate rows and tweets 2. Incorporate mentor review feedback 3. Extract the positive , negative and neutral sentiment from raw tweets using TextBlob 4. Remove rows from a data frame (merge\_tweets\_df) that are identical (tweet\_id) to filtered\_week\_positive 5. Review with the mentor 6. Incorporate mentor review feedback 7. Cleaning the data 8. Convert sentiment\_category to numeric class for machine learning. 9. Visualizing the Data 10. Preparation for mid-review (PPT/Code Review) 11. Review with mentor 12. Mid Viva | Completed |
| Week# 5 & 6 | * Explore the SKLearn, TF-IDF Scikit-Learn Library and Ensemble Model performance. * Explore the TfidfVectorizer for feature extraction. * Explore the Various Models (RandomForestClassifier, GradientBoostingClassifier and SVM) for Multiclass classification. * Explore various Models (LogisticRegression, DecisionTreeClassifier, RandomForestClassifier, GaussianNB, AdaBoostClassifier, GradientBoostingClassifier, KNeighborsClassifier, XGBClassifier, LinearSVC, BalancedRandomForestClassifier, BalancedBaggingClassifier and EasyEnsembleClassifier) for Binary classification. * Create a PPT to provide updates to Reviewer during the bi-weekly review. | Completed |
| Week# 7 & 8 | * Generate a spreadsheet with the comparison statistics between the models. * Finalize the top-2 models with best performance across SKLearn and Ensemble Models. * Prepare the project report using the template provided by incorporating the results of all the models explored for this project. Also, include the additional information as per the template. * Review the project progress with SME during the sync-ups and incorporate the suggestions. * Create PPT for the final project review. * Preparation & submission of Solution * Review of solution * Viva Voce | Completed |

## Approach

**Data Collection and Preprocessing**: The project initiated with data acquisition from Gautam Gangopadhyay, who provided the dataset in CSV format. This dataset comprised tweets from airline passengers but lacked sentiment labels. Preprocessing ensued to determine sentiment labels for each tweet. Utilizing sentiment analysis tools such as Vader and TextBlob, sentiment scores were assigned to the text inputs, inferring the sentiment polarity (positive, negative, neutral) of each tweet.

**Model Selection and Hyperparameter Tuning**: Following sentiment labeling, the project focused on selecting suitable machine learning algorithms for sentiment analysis. A diverse array of classifiers including Logistic Regression, Decision Trees, Random Forests, Naive Bayes, Boosting algorithms, K-Nearest Neighbors, Extreme Gradient Boosting, Support Vector Machines, and possibly deep learning models were considered. Furthermore, hyperparameters underwent systematic tuning to optimize performance. Techniques such as grid search or random search were utilized to explore hyperparameter combinations. Cross-validation methods such as k-fold cross-validation were employed to assess model performance across data subsets. Evaluation metrics encompassing accuracy, precision, recall, F1-score, and ROC AUC score were utilized to comprehensively evaluate model performance, facilitating the selection of the optimal model for sentiment analysis of airline passengers' tweets.

**Deployment**: Upon selecting the optimal sentiment analysis model, the next phase involved deployment into a production environment. Integration into a user-friendly interface, such as a web application, was likely considered to allow stakeholders easy access to the sentiment analysis tool. This facilitated real-time monitoring and analysis of customer feedback for airlines.

**Continuous Improvement**: The solution was not static; instead, it underwent continuous monitoring and refinement to ensure optimal performance over time. This entailed periodic retraining of the model with updated data, fine-tuning of model parameters, and adaptation to evolving sentiment patterns. Feedback mechanisms were also implemented to gather insights from end-users and stakeholders, guiding iterative enhancements to the sentiment analysis solution.

By adhering to this holistic approach, the project aimed to develop an efficient, scalable, and adaptable solution for sentiment analysis, furnishing invaluable insights to airlines for augmenting customer satisfaction and refining service quality.

# **Model Development**

## Feature Engineering / Data Pre-processing activities

Feature Engineering activities performed:

* Read the data from the Input File
* Finding the missing values in Each Column
* Creating the Label Columns from Tweet ID
* Dropping rows with NULL tweets
* Drop duplicate rows and tweet ID
* Filling NULL tweet source and tweet by columns
* Dropping Date Column
* Cleaning Tweet\_Source column to follow same format
* Visualizing the data as PIE Charts
* Visualizing the data as Word Cloud to see patterns.

## Generating Machine Learning Data models

* Sentiments were added to the words using Vader after completing data preprocessing and feature engineering.
* Lemmatization was applied to standardize the word forms in the tweets data.
* Non-proper English words were removed using the NLTK library.
* Vectors of the words were obtained using the TfidfVectorizer.
* An array containing all the models was created, and their performance was evaluated on the data generated in the previous step.
* The metrics obtained were stored in a list and sorted to identify the top three performing models.
* Hyperparameter tuning was performed on these three models to optimize their performance further.
* The best model along with its optimal parameters was identified.
* Finally, the model was exported to a .pkl file for future use in the application.

## Libraries

Following are the libraries imported for developing the various data models built as part of this project.

* os
* sys
* scipy
* scipy.stats
* numpy
* pandas
* matplotlib
* matplotlib.pyplot

**sklearn**

* sklearn.ensemble.RandomForestClassifier
* sklearn.model\_selection.train\_test\_split
* sklearn.model\_selection.GridSearchCV
* sklearn.model\_selection.StratifiedKFold,
* sklearn.model\_selection.cross\_val\_score
* sklearn.model\_selection.cross\_val\_predict
* sklearn.preprocessing.StandardScaler
* sklearn.decomposition.PCA
* sklearn.metrics.confusion\_matrix
* sklearn.metrics.accuracy\_score
* sklearn.metrics.classification\_report
* sklearn.metrics.precision\_score
* sklearn.metrics.recall\_score
* sklearn.metrics.roc\_auc\_score
* sklearn.metrics.roc\_curve
* sklearn.metrics.f1\_score
* sklearn.svm.SVC
* sklearn.tree.DecisionTreeClassifier
* imblearn
* imblearn.over\_sampling.SMOTE
* xgboost
* collections.Counter
* itertools

**Apache Airflow Dependencies:**

* apache-airflow
* apache-airflow[postgres] (if using PostgreSQL as the metadata database)
* apache-airflow[celery] (if using CeleryExecutor)
* apache-airflow[async] (if using AsyncExecutor)
* apache-airflow[aws] (for AWS dependencies)
* apache-airflow[google] (for Google Cloud dependencies)
* apache-airflow[microsoft.azure] (for Microsoft Azure dependencies)
* apache-airflow[kubernetes] (for Kubernetes dependencies)

**Streamlit Dependencies:**

* streamlit

## Test/Train Split

Before building any data model, the project data is split into training (80%) and test data (20%). Stratify parameter is applied on Label parameter so that the Label data is distributed evenly across training and test data. Random parameter is set to 0 This parameter ensures reproducibility by fixing the random seed for the split. It ensures that the same split is obtained every time the code is executed, which is crucial for consistency in model evaluation and comparison.

## SKLearn Models

* SKLearn models for multiclass classification
* SKLearn models for binary classification
* parameter tuning for the models, identifying the parameters optimized and determining the best-performing model
* deep learning techniques for further model enhancement

## SKLearn models for multiclass classification

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR LogisticRegression():

ACCURACY: 0.8864734299516909

F1 SCORE: 0.9083820662768031

CONFUSION MATRIX:

[[ 670 148]

[ 87 1165]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.89 0.82 0.85 818

1 0.89 0.93 0.91 1252

accuracy 0.89 2070

macro avg 0.89 0.87 0.88 2070

weighted avg 0.89 0.89 0.89 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.8747910433770514

A line graph with orange and black lines

Description automatically generated

Total time taken to run the model : 2.391084671020508 seconds

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR DecisionTreeClassifier():

ACCURACY: 0.8077294685990338

F1 SCORE: 0.8382113821138211

CONFUSION MATRIX:

[[ 641 177]

[ 221 1031]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.74 0.78 0.76 818

1 0.85 0.82 0.84 1252

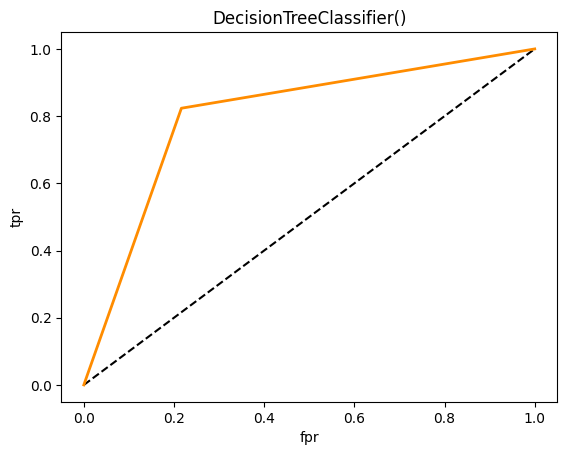
accuracy 0.81 2070

macro avg 0.80 0.80 0.80 2070

weighted avg 0.81 0.81 0.81 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.8035505050110532



Total time taken to run the model : 13.19615626335144 seconds

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR RandomForestClassifier():

ACCURACY: 0.8647342995169082

F1 SCORE: 0.8896769109535068

CONFUSION MATRIX:

[[ 661 157]

[ 123 1129]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.84 0.81 0.83 818

1 0.88 0.90 0.89 1252

accuracy 0.86 2070

macro avg 0.86 0.85 0.86 2070

weighted avg 0.86 0.86 0.86 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.8549128240780521

A line graph with orange and black lines

Description automatically generated

Total time taken to run the model : 29.56759214401245 seconds

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR GaussianNB():

ACCURACY: 0.5183574879227053

F1 SCORE: 0.4179801517805021

CONFUSION MATRIX:

[[715 103]

[894 358]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.44 0.87 0.59 818

1 0.78 0.29 0.42 1252

accuracy 0.52 2070

macro avg 0.61 0.58 0.50 2070

weighted avg 0.65 0.52 0.49 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.5800128107985658

A line graph with orange line

Description automatically generated

Total time taken to run the model : 1.766416072845459 seconds

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR AdaBoostClassifier():

ACCURACY: 0.8072463768115942

F1 SCORE: 0.8547506370586094

CONFUSION MATRIX:

[[ 497 321]

[ 78 1174]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.86 0.61 0.71 818

1 0.79 0.94 0.85 1252

accuracy 0.81 2070

macro avg 0.82 0.77 0.78 2070

weighted avg 0.82 0.81 0.80 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.7726395713069358

A line graph with orange and black lines

Description automatically generated

Total time taken to run the model : 38.189204454422 seconds

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR GradientBoostingClassifier():

ACCURACY: 0.8164251207729468

F1 SCORE: 0.8617176128093159

CONFUSION MATRIX:

[[ 506 312]

[ 68 1184]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.88 0.62 0.73 818

1 0.79 0.95 0.86 1252

accuracy 0.82 2070

macro avg 0.84 0.78 0.79 2070

weighted avg 0.83 0.82 0.81 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.7821344040244654

A line graph with orange and black lines

Description automatically generated

Total time taken to run the model : 134.49189829826355 seconds

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR KNeighborsClassifier():

ACCURACY: 0.5666666666666667

F1 SCORE: 0.4701712935617248

CONFUSION MATRIX:

[[775 43]

[854 398]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.48 0.95 0.63 818

1 0.90 0.32 0.47 1252

accuracy 0.57 2070

macro avg 0.69 0.63 0.55 2070

weighted avg 0.73 0.57 0.53 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.6326620683190515

A line graph with orange and black lines

Description automatically generated

Total time taken to run the model : 26.683622121810913 seconds

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR XGBClassifier(base\_score=None, booster=None, callbacks=None,

colsample\_bylevel=None, colsample\_bynode=None,

colsample\_bytree=None, device=None, early\_stopping\_rounds=None,

enable\_categorical=False, eval\_metric=None, feature\_types=None,

gamma=None, grow\_policy=None, importance\_type=None,

interaction\_constraints=None, learning\_rate=None, max\_bin=None,

max\_cat\_threshold=None, max\_cat\_to\_onehot=None,

max\_delta\_step=None, max\_depth=None, max\_leaves=None,

min\_child\_weight=None, missing=nan, monotone\_constraints=None,

multi\_strategy=None, n\_estimators=None, n\_jobs=None,

num\_parallel\_tree=None, random\_state=None, ...):

ACCURACY: 0.866183574879227

F1 SCORE: 0.8924271844660194

CONFUSION MATRIX:

[[ 644 174]

[ 103 1149]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.86 0.79 0.82 818

1 0.87 0.92 0.89 1252

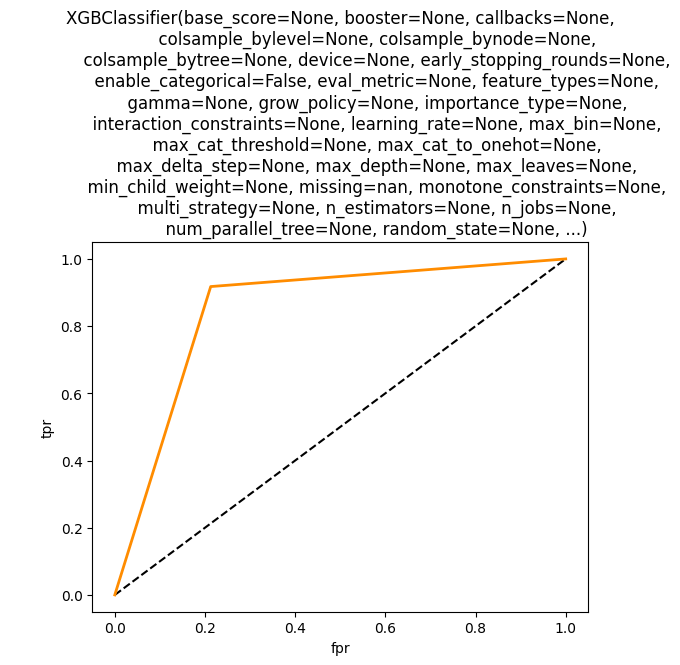
accuracy 0.87 2070

macro avg 0.87 0.85 0.86 2070

weighted avg 0.87 0.87 0.86 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.8525088464813267



Total time taken to run the model : 12.247537612915039 seconds

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FOR LinearSVC():

ACCURACY: 0.9115942028985508

F1 SCORE: 0.9274097580325267

CONFUSION MATRIX:

[[ 718 100]

[ 83 1169]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.90 0.88 0.89 818

1 0.92 0.93 0.93 1252

accuracy 0.91 2070

macro avg 0.91 0.91 0.91 2070

weighted avg 0.91 0.91 0.91 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.9057283407672418

A graph with a line and a point

Description automatically generated with medium confidence

Total time taken to run the model : 0.5492215156555176 seconds

CPU times: user 4min 36s, sys: 7.18 s, total: 4min 43s

Wall time: 4min 19s

## SKLearn models for Binary classification

FOR LinearSVC(max\_iter=40000):

ACCURACY: 0.9115942028985508

F1 SCORE: 0.9274097580325267

CONFUSION MATRIX:

[[ 718 100]

[ 83 1169]]

CLASSIFICATION REPORT:

precision recall f1-score support

0 0.90 0.88 0.89 818

1 0.92 0.93 0.93 1252

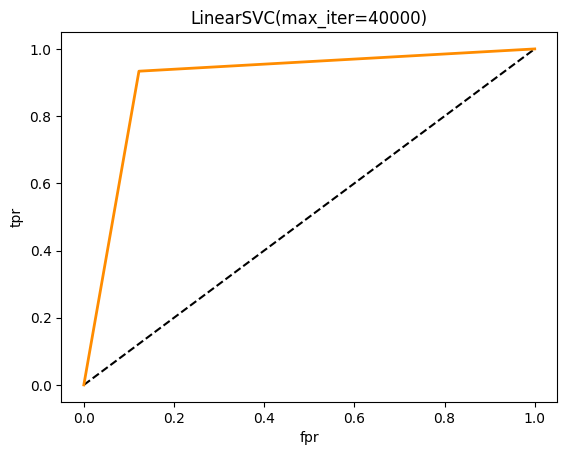
accuracy 0.91 2070

macro avg 0.91 0.91 0.91 2070

weighted avg 0.91 0.91 0.91 2070

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AUC roc\_auc\_score: 0.9057283407672418



Total time taken to run the model : 0.5314600467681885 seconds

## SKLearn models for Binary classification – Hyper Parameter Tuning

LinearSVC

Best Hyperparameters: {'C': 1, 'loss': 'squared\_hinge', 'max\_iter': 40000}

Test Accuracy: 0.9115942028985508

LogisticRegression

Best Hyperparameters: {'C': 10, 'max\_iter': 40000, 'penalty': 'l2'}

Test Accuracy: 0.908695652173913

RandomForestClassifier

Best Hyperparameters: {'criterion': 'entropy'}

Test Accuracy: 0.8690821256038648

# **Conclusions / Recommendations**

Following is the performance summary of the various SKLEARN models implemented as part of this project.

**SKLearn Models Comparison Report**

| **Model** | **Accuracy** | **F1 Score** | **AUC roc\_auc\_score** | **Total Time Taken (seconds)** |
| --- | --- | --- | --- | --- |
| Logistic Regression | 0.8865 | 0.9084 | 0.8748 | 2.39 |
| Decision Tree | 0.8077 | 0.8382 | 0.8036 | 13.20 |
| Random Forest | 0.8647 | 0.8897 | 0.8549 | 29.57 |
| Gaussian Naive Bayes | 0.5184 | 0.4180 | 0.5800 | 1.77 |
| AdaBoost | 0.8072 | 0.8548 | 0.7726 | 38.19 |
| Gradient Boosting | 0.8164 | 0.8617 | 0.7821 | 134.49 |
| K-Nearest Neighbors | 0.5667 | 0.4702 | 0.6327 | 26.68 |
| XGBoost | 0.8662 | 0.8924 | 0.8525 | 12.25 |
| Linear SVC | 0.9116 | 0.9274 | 0.9057 | 0.55 |

Note: Neutral sentiment is not taken into consideration in the evaluation metrics.

This comprehensive summary highlights the previous work conducted in sentiment analysis, along with the model performance metrics and the acknowledgment that neutral sentiment was not factored into the analysis.

Our primary objective was to accurately predict negative tweets, specifically aiming to minimize False Negatives. Therefore, utilizing Recall as our evaluation metric, the following models demonstrated the best performance:

* LinearSVC
* LogisticRegression
* RandomForestClassifier

# **Future Work & Extension or Scope of improvements**

1. We have limited number of raw data., if we get more data then we can train the model more efficiently.

2. In this data we should get balanced class distribution

3. Preprocessing of tweets – there are lot of scope to perform the preprocessing the tweets but we have done limited steps because of time

4. Parameter tuning – each algorithm we have lot of parameters to tune the model but because of time consuming we have done parameter tuning only on 2 algorithms and used few parameters.

5. We have to explore in detail NN , DL , LSTM

6. We have to check BERT(Bidirectional Encoder Representations from Transformers) language model is an open source machine learning framework for natural language processing (NLP) for better accuracy.

7. Polarity of each sentiment(weak positive and weak negative)

8. Create a Chrome/ Firefox Plugin + Telegram / WhatsApp Bot

9. Sentiment based news.

# **Bibliography / References**

Referred to following websites for learning the machine learning models as well as usage of various technical parameters for fine tuning the models.

* <https://www.analyticsvidhya.com/>
* <https://towardsdatascience.com/>
* <https://scikit-learn.org/>
* <https://medium.com/>
* <https://towardsdatascience.com/evaluation-metrics-for-classification-problems-in-machine-learning-d9f9c7313190>

# **Appendices**

**Code Base:**

Enclosed the machine-learning and deep-learning code base developed as part of this project. This code is written in Python and would need basic as well as specific Python Libraries related to various models developed for this project. Please install all those libraries before executing the project. Also, this code needs to be run in GPU processing environment with for effective performance.

Following are the notebook files we uploaded to Canvas.

* Pre-processing Notebook
  + PCAM ZC321-C9-CODE-Sentiment Analysis\_Data\_Understanding-Gr-5-Step-1.ipynb
* Sentiment Analysis
  + PCAM ZC321-C9-CODE-Sentiment Analysis\_DataCleaning\_SentimentCreation-Gr-5-Step-2.ipynb
* Model Creation
  + PCAM ZC321-C9-CODE-Sentiment Analysis\_Main\_Model\_Selection-Gr-5-Step-3.ipynb.ipynb
* Deep Learning Model Creation
  + PCAM ZC321-C9-CODE-Sentiment Analysis\_Model\_DeepLearning\_NN-Gr-5-Step-4.ipynb
* Streamlit app:
  + streamlitApp.py
* Airflow DAG
  + TweetModel\_Recreate.py

**Model Comparison Reports:**

* We have only one csv that need to mentioned.

**Check list of items for the Final report**

1. Is the Cover page in proper format? Y
2. Is the Title page in proper format? Y
3. Is the Certificate from the Mentor in proper format? Has it been signed? Y
4. Is Abstract included in the Report? Is it properly written? N
5. Does the Table of Contents page include chapter page numbers? N
6. Does the Report contain a summary of the literature survey? N
   1. Are the Pages numbered properly? N
   2. Are the Figures numbered properly? N
   3. Are the Tables numbered properly? N
   4. Are the Captions for the Figures and Tables proper? N
   5. Are the Appendices numbered? N
      1. Does the Report have Conclusion / Recommendations of the work? N
      2. Are References/Bibliography given in the Report? N
      3. Have the References been cited in the Report? N
      4. Is the citation of References / Bibliography in proper format? N