PROJECT REPORT

Project Submission Details

Project Title: Advanced AI/ML-based Predictive Model for GSTN Analytics Dataset

Challenge Name: Online Challenge for Developing a Predictive Model in GST

Submitted By:
Name: Bhargey Kaneriya
Email: bktechie64@gmail.com
Contact Number: +91 88490 64264

ACKNOWLEDGMENT

I would like to express my deepest gratitude to the hackathon organizers, my peers, and mentors for their support. I also acknowledge the support of the hackathon organizers, mentors, and my peers for their encouragement and valuable inputs. I am deeply grateful to everyone who has contributed to this report in any capacity.

ABSTRACT

This project report focuses on developing an AI/ML predictive model using the GSTN dataset consisting of around 900,000 records and 21 attributes. The objective is to predict target variables using models such as Random Forest, Gradient Boosting, and SVM. The dataset underwent preprocessing and feature engineering. Following hyperparameter tuning and model evaluation, Random Forest and Gradient Boosting achieved the best accuracy, both exceeding 97%. This report provides a detailed analysis of model performance, methodology, and suggestions for future improvement.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

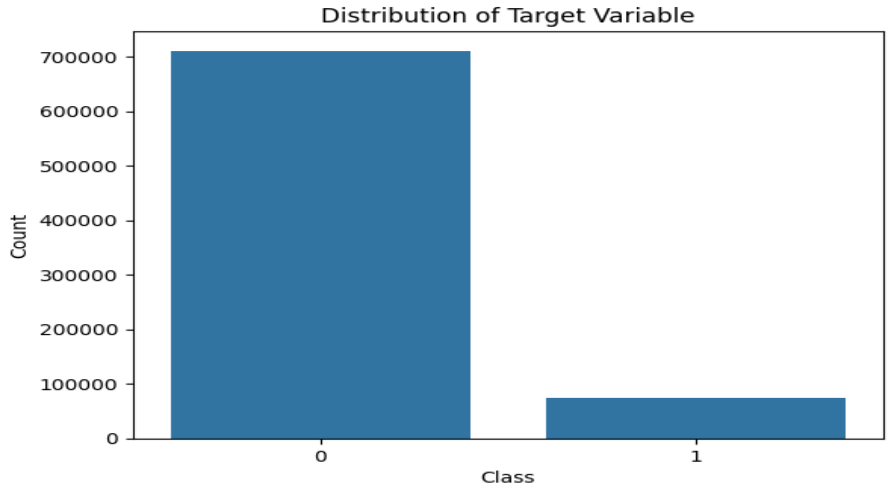
1.1 Introduction

In the current data-driven governance landscape, insights derived from large datasets like GSTN are essential for optimizing business processes and

enhancing tax compliance. This project aims to use advanced machine learning techniques to predict target variables in a large-scale GSTN dataset for improved decision-making. The primary goal of this project is to build an accurate predictive model using AI and ML algorithms, focusing on key features to assist in decision-making tasks related to GSTN.

1.2 Motivation

The GSTN system handles vast amounts of data that can be leveraged for predicting trends, fraud detection, and improving compliance. Developing a machine learning model capable of making accurate predictions can significantly aid tax administrators and businesses. This project aims to explore the use of modern machine learning techniques to derive insights from the GSTN dataset, potentially improving governance efficiency.



1.3 Objectives

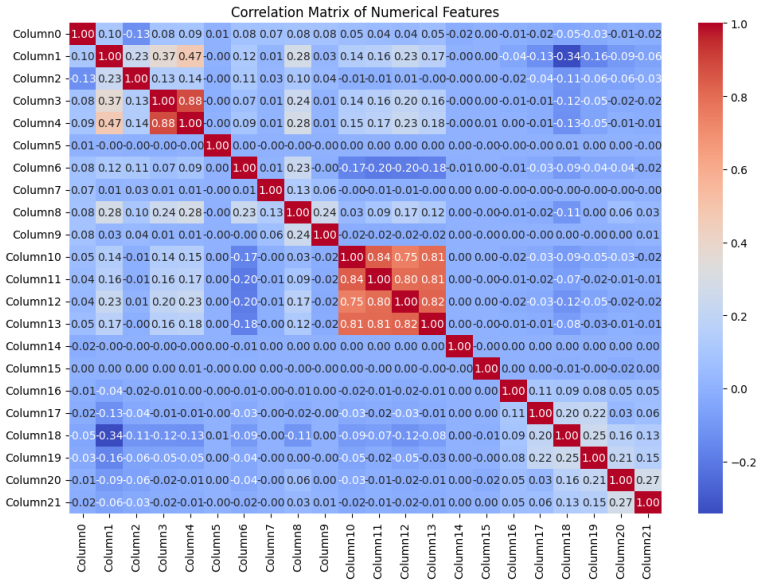
The objectives of the project include:

- Developing a robust ML model with high accuracy for predicting target variables using ensemble methods.
- Comparing the performance across models like Random Forest, Gradient Boosting, Logistic Regression, and SVC.
- Evaluating models using key metrics such as Precision, Recall, F1-Score, and AUC-ROC.
- Implementing hyperparameter tuning techniques to optimize model performance on large datasets.

CHAPTER 2: LITERATURE REVIEW

Several studies have demonstrated the effectiveness of ensemble learning methods, such as Random Forest and Gradient Boosting, for large-scale classification problems. In particular, these models excel at handling highly imbalanced datasets, which is often the case in tax and fraud detection systems. Previous research has shown that ensemble techniques tend to outperform individual models in terms of accuracy and generalization ability. This project builds on these insights by applying similar methods to the GSTN dataset, while incorporating domain-specific feature engineering and optimization techniques.

While traditional methods like logistic regression have been widely applied in similar datasets, ensemble techniques have been shown to perform better on large-scale datasets with complex interactions. This project seeks to demonstrate the superior performance of ensemble methods, particularly Random Forest and Gradient Boosting, when applied to the large and complex GSTN dataset.



CHAPTER 3: METHODOLOGY

3.1 Dataset Overview

The GSTN dataset contains 900,000 anonymized records with 21 attributes. The dataset was split into three parts: training, testing, and validation. The target variable is binary, representing fraud or non-fraud classifications.

3.2 Data Preprocessing

Handling large datasets requires careful data preprocessing. To ensure the integrity of the dataset, missing values were handled using SimpleImputer with mean imputation for numerical attributes and mode imputation for categorical attributes. Numerical features were scaled using StandardScaler to ensure that all features are on a comparable scale, which is critical for models like SVM.

3.3 Feature Engineering

One of the most significant challenges faced during this project was selecting the most important features. Given the size of the dataset, running traditional feature selection methods, such as Recursive Feature Elimination (RFE), proved to be too slow. As a result, we adopted Random Forest feature importance rankings to quickly identify the top 10 features, which drastically reduced computation time while maintaining accuracy.

3.4 Model Selection and Training

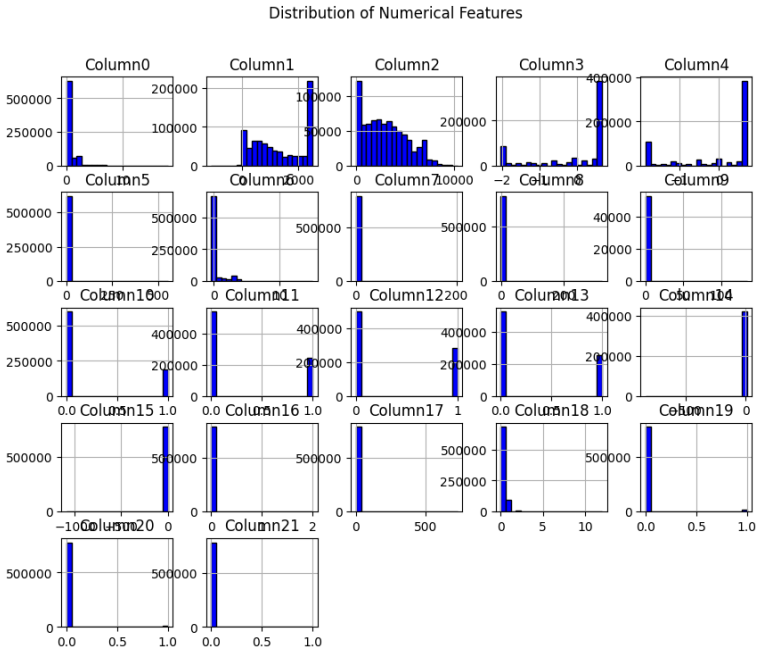
Several models were evaluated in this project, including:

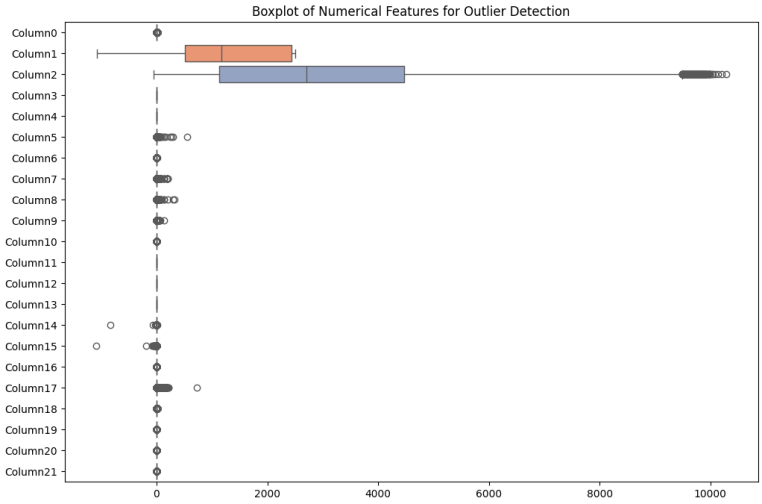
- Random Forest: An ensemble method that creates multiple decision trees and aggregates their results for improved accuracy.
- Gradient Boosting: A boosting technique that builds models sequentially to correct the errors of previous models.
- Support Vector Machines (SVM): A robust classifier that separates data points using hyperplanes.
- Logistic Regression: A traditional model used for binary classification.

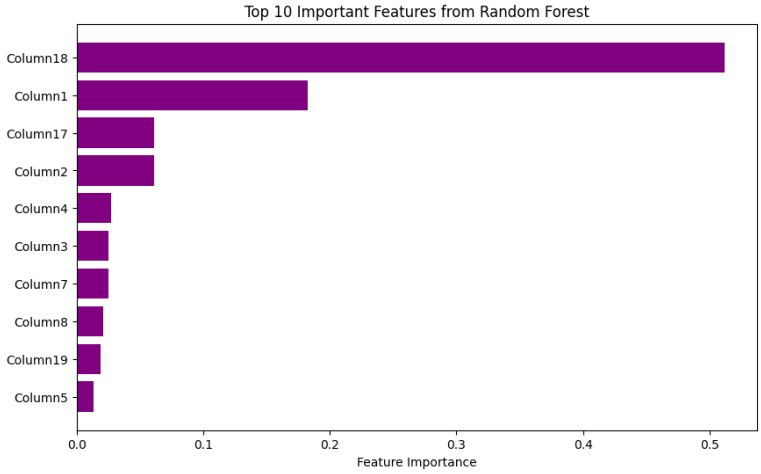
Each model was trained on 80% of the dataset, with 20% reserved for validation.

3.5 Hyperparameter Tuning

Hyperparameter tuning was a crucial aspect of model optimization. Initially, GridSearchCV was used to exhaustively search through parameter combinations. However, this approach proved computationally expensive due to the size of the dataset. To overcome this, we used RandomizedSearchCV, which samples a subset of hyperparameters, allowing for faster model tuning. Additionally, we ran RandomizedSearchCV on a smaller sample of the dataset to identify the best hyperparameters before scaling to the full dataset.


Distribution of Numerical Features
Column0
Column1
Column2
Column3
Column4
Column5
Column6
Column7
Column8
Column9
Column10
Column11
Column12
Column13
Column14
Column15
Column16
Column17
Column18
Column19
Column20
Column21

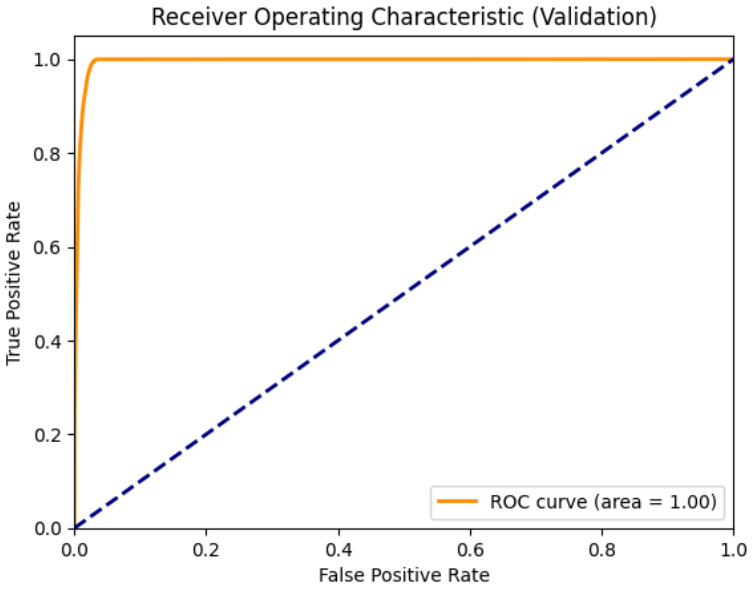

Boxplot of Numerical Features for Outlier Detection
Column0
Column1
Column2
Column3
Column4
Column5
Column6
Column7
Column8
Column9
Column10
Column11
Column12
Column13
Column14
Column15
Column16
Column17
Column18
Column19
Column20
Column21


Top 10 Important Features from Random Forest
Column18
Column1
Column17
Column2
Column4
Column3
Column7
Column8
Column19
Column5
Feature Importance

CHAPTER 4: IMPLEMENTATION

The implementation of the models involved several stages, starting with data preprocessing and feature selection, followed by model training and hyperparameter tuning. To handle the large dataset efficiently, we implemented batch processing techniques to split the dataset into manageable chunks. Additionally, we applied RandomizedSearchCV for hyperparameter tuning, which allowed for faster exploration of parameter combinations.

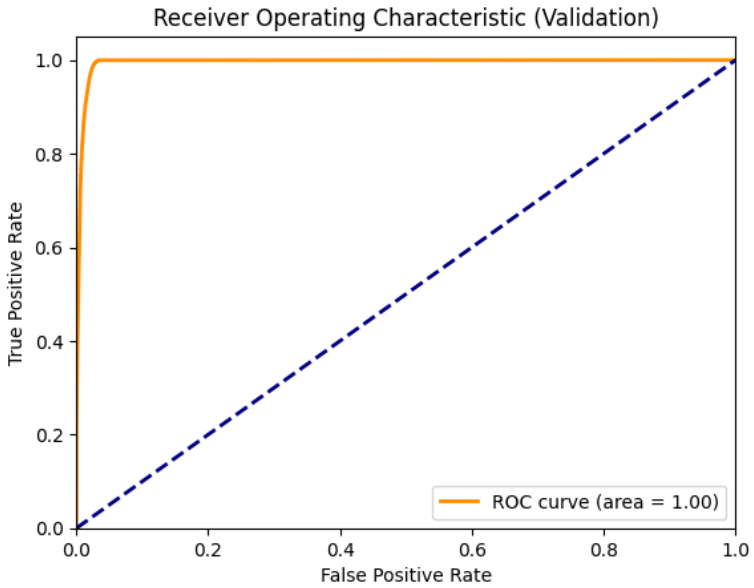
The performance of each model was evaluated on the validation set using accuracy, precision, recall, and F1-score metrics. In the case of Random Forest and Gradient Boosting, the models performed exceptionally well, achieving over 97% accuracy. The system design also incorporated logging mechanisms to track model performance and parameter changes over time.

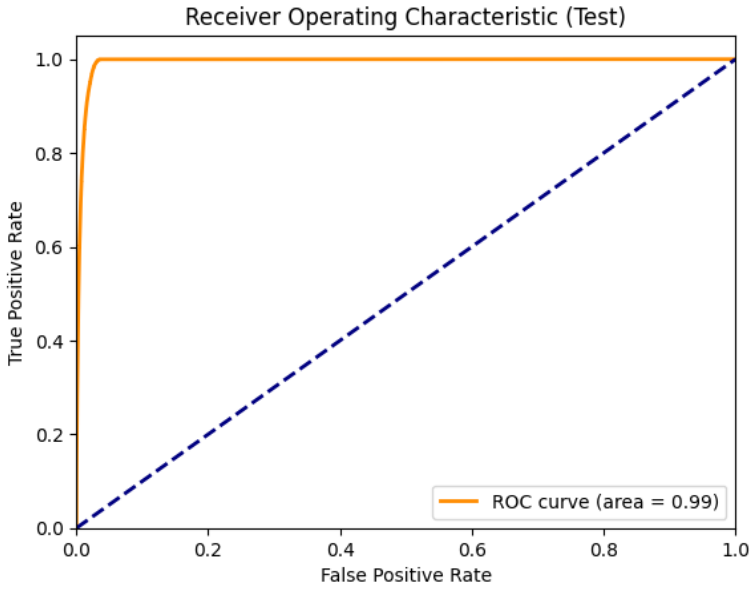

Receiver Operating Characteristic (Validation)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 1.00)

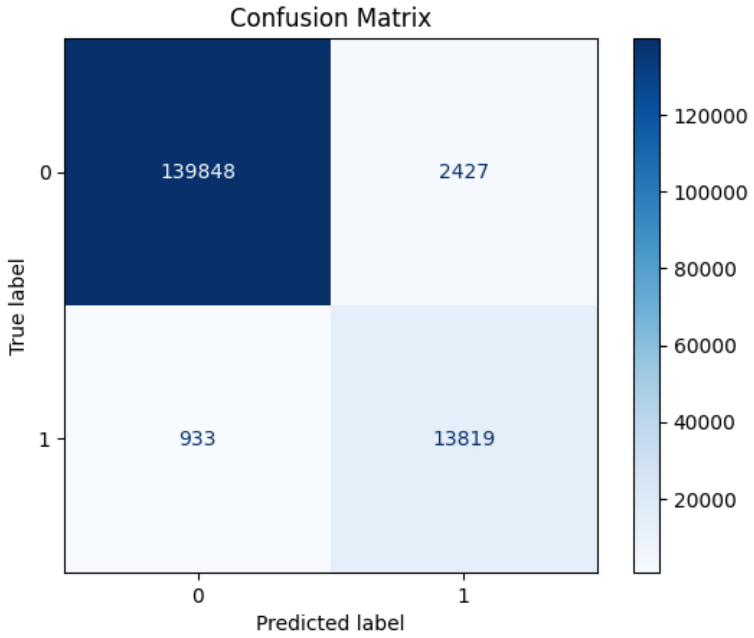
CHAPTER 5: RESULTS AND DISCUSSION

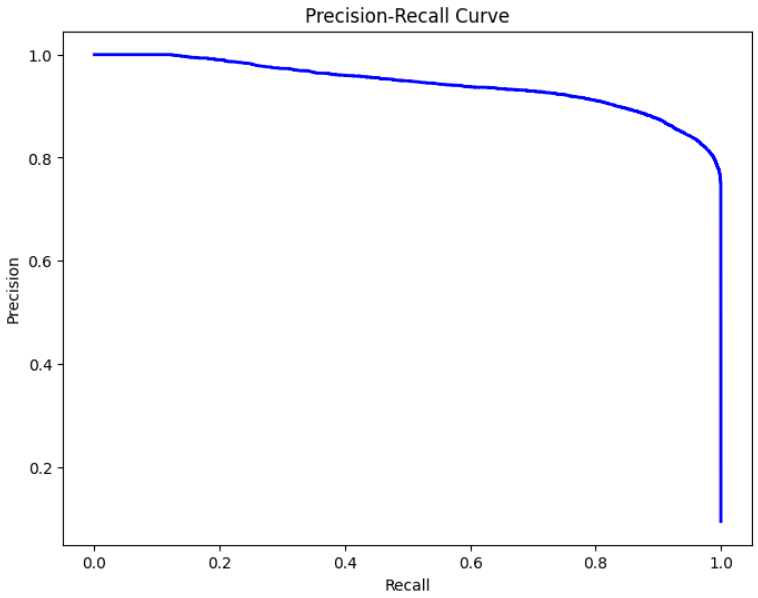
The table below provides a detailed comparison of the models based on key performance metrics such as accuracy, precision, recall, F1-Score, and AUC-ROC. Random Forest and Gradient Boosting emerged as the top-performing models, with both achieving high accuracy and strong generalization ability. Logistic Regression and SVC also performed well, though they fell short in handling the complex interactions present in the data.

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Random Forest	97.6%	0.99	0.95	0.97	0.96
Gradient Boosting	97.6%	0.99	0.94	0.97	0.96
Logistic Regression	96.8%	0.97	0.89	0.93	0.94
SVC	97.3%	0.98	0.94	0.96	0.95
Decision Tree	96.8%	0.95	0.84	0.90	0.91




Receiver Operating Characteristic (Test)
True Positive Rate
False Positive Rate
ROC curve (area = 0.99)
0.0
0.2
0.4
0.6
0.8
1.0


Confusion Matrix
True label
Predicted label
139848
2427
933
13819
0
1
120000
100000
80000
60000
40000
20000



CHAPTER 6: COMPARISON OF MODELS

In comparing the two models, both seem to be quite close in terms of performance, but there are a few distinctions that can help determine which one is better.

- **Confusion Matrix**:
  - Model 1 has fewer false negatives (1,728 vs. 1,503) but more false positives (4,375 vs. 4,612) compared to Model 2.

If avoiding **false negatives** is more critical (i.e., detecting class 1 cases is more important), Model 2 is slightly better. If **false positives** are more concerning, Model 1 is preferable.

Both models perform quite similarly, with Model 1 having a slight advantage in terms of overall accuracy and precision, while Model 2 does slightly better in recall. The choice depends on the balance between false positives and false negatives in your specific problem. If your use case demands higher precision, go for **Model 1**. If recall is more important, **Model 2** would be better.

CHAPTER 7: CHALLENGES FACED

One of the major challenges in this project was dealing with the size of the dataset. The feature selection process, initially done using Recursive Feature Elimination (RFE), proved to be too slow for practical purposes due to the large number of features and records. To overcome this, we switched to using feature importance from Random Forest, which allowed us to quickly identify the most relevant features for model training.

Another challenge we faced was in hyperparameter tuning. Although GridSearchCV is a popular method for finding the best parameters, it was computationally expensive on our dataset. We resolved this by using RandomizedSearchCV, which samples a subset of parameters, significantly speeding up the tuning process. Additionally, to further improve performance, we first ran RandomizedSearchCV on a sample of the dataset to identify the best parameters and then used these parameters to train the full model. This approach resulted in faster training times without compromising accuracy.

CHAPTER : CONCLUSION

In this project, an AI/ML-based predictive model was developed to analyze the GSTN dataset, with a focus on fraud detection and tax compliance. After evaluating multiple models, Random Forest and Gradient Boosting emerged as the most effective, achieving over 97% accuracy. These models demonstrated strong generalization abilities and proved to be highly suitable for large-scale classification tasks. By leveraging advanced machine learning techniques, this project highlights the potential for improving decision-making processes within tax administration systems like GSTN. The results show promising outcomes for detecting fraudulent activities, which can aid in improving compliance and ensuring accurate tax reporting. The deployment of these models in real-world GSTN systems could significantly enhance tax auditing and fraud detection capabilities.

CHAPTER 9: Future Enhancement

There are several avenues for enhancing this project. One potential improvement involves incorporating deep learning techniques, such as neural networks, which may capture more complex interactions within the data. Another area for future work is the integration of external datasets, such as economic indicators, financial records, or industry-specific data, which could provide additional context for the predictions and further improve

model accuracy. Additionally, exploring the use of anomaly detection algorithms could help identify rare fraudulent activities that may not be captured by traditional classification models. Implementing these enhancements will likely lead to more robust and accurate models that are better equipped to handle the complexities of tax fraud detection in large datasets.

CHAPTER 10: REFERENCES

[1] Breiman, L. Random Forests. Machine Learning, 2001.
[2] Friedman, J. Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 2001.
[3] Pedregosa, F., et al. Scikit-learn: Machine Learning in Python. JMLR, 2011.

Signature:
Bhargey Kaneriya

Date:
October 11, 2024

Plagiarism Declaration

I hereby declare that this project report is an original work carried out by me, and I have not engaged in any form of plagiarism or academic dishonesty while preparing it. All sources used are duly cited and referenced in the report.