ANTICIPATORY VEHICLE SERVICING

FORECASTING MAINTENANCE

NEEDS

MINI PROJECT REPORT

Submitted by

HARI AMERTHESH N	**210701067**
KRISHNAKUMAR R	**210701126**
KISHORE KARTHIK M	**210701123**

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY:: CHENNAI 600 025

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**Anticipatory Vehicle Servicing Forecasting Maintenance**" is the bonafide work of **"Hari Amerthesh N (210701067) , Krishnakumar R (210701126) ,and Kishore Karthik M (210701123)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. K. ANAND, ME , Ph.D.
Professor,
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Thandalam
Chennai – 602105

Submitted to Mini Project Viva-Voce Examination held on ____________________

Internal Examiner **External Examiner**

ABSTRACT

In Recent days the Advanced technologies called predictive vehicle maintenance systems (PVMS) enable vehicle owners and operators to identify possible issues with their cars before they arise. These systems track the operation of vital auto parts including engines, brakes, and transmissions using data from numerous sensors and other sources. The owner or operator can take preventive action by analyzing this data to find trends that point to the likelihood of a failure or other problem. Sensors, data processing software, and decision-making algorithms are the three primary parts of PVMS. The sensors gather information on a range of performance metrics for the car, including tyre pressure, oil pressure, and engine temperature. After processing this data, software inspects it for indications of potential problems, employing algorithms that account for things like the vehicle's age and condition, the weather, and the style of operation. When a problem is found, the system can notify the driver or owner of the car, giving them a heads-up and offering a suggestion for what to do. In some circumstances, the system may even initiate remedial action on its own, such as modifying engine performance to avoid overheating or cut down on fuel usage. Compared to conventional methods of vehicle maintenance, PVMS have a number of benefits. They can aid in lowering the possibility of unforeseen breakdowns, which can be expensive and inconvenient, particularly for commercial vehicles. They can also aid in extending the life of essential components, lowering the need for pricey repairs and replacements. They can also contribute to increased safety by notifying drivers of potential problems that might affect how well their cars perform. Despite their advantages, PVMS have certain drawbacks. The necessity for correct data, which can occasionally be difficult to obtain, is one of the main obstacles. It can also be expensive to establish these systems, particularly for smaller operations or individual vehicle owners. However, it is anticipated that PVMS will become a more crucial tool for vehicle operation.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Dr. K. ANAND ME, Ph.D.** Assistant Professor, Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

HARI AMERTHESH N -210701067
KRISHNAKUMAR M -210701126
KISHORE KARTHIK M -210701123

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 GENERAL

In the Globally connected world of vehicle predictive maintenance system, the presence of comment toxicity has become a major concern. However, with everything having a set of drawbacks, exists a growing issue of toxic feedback, resulting in negative impacts by affecting the mental health of people.

1.2 OBJECTIVE

In the ever-changing world vehicle maintenance system, the proliferation of platforms, forums, and discussion threads has resulted in a flood of user-generated content. Toxic remarks including hostility, anger, or inflammatory language have the potential to disrupt online conversation .To address this issue, our project uses advanced computational techniques such as neural networks, random forests, and natural language processing (NLP) to classify comments according to their toxicity levels.

1.3 EXISTING SYSTEM

Several existing methods in machine learning projects use natural language processing (NLP) techniques such as Naive Bayes and TF-IDF to analyze harmful comments. Jigsaw, an Alphabet Inc. company, created the Perspective API, which is a significant example. This API utilizes machine learning models to assess the toxicity of comments and assigns a toxicity score ranging from 0 to 1. It classifies toxicity into six types: toxic, severe toxic, obscene, threat, insult, and identity hate.

1.3 PROPOSED SYSTEM

We hope to automate the time-consuming task of assessing large amounts of textual feedback from many sources and situations by employing machine learning methods. From consumer reviews and social media comments to employee surveys and product evaluations, this project covers the entire spectrum of input, shedding light on the trends and feelings that influence decision-making. The overarching goal is to give companies the opportunity to generate effective suggestions from feedback data.

CHAPTER 2
LITERATURE SURVEY

[1] In order to identify toxicity in textual comments, we evaluated several deep learning algorithms fed by different word embedding representations in this paper. We can categorically conclude from the results that toxicity can be detected using methods such as deep learning and machine learning that are given syntactic and semantic data that has been collected from the text. We demonstrate that, out of the tested models, the LSTM-based approach is the most preferred one for toxicity detection. We also demonstrate how different word embeddings can represent the domain's expertise in different ways, and that having a single model that works in every situation might not be sufficient. The results are particularly promising when imitating approaches are applied to OOV words, for which there are insufficient examples to construct meaningful domain-dependent word embeddings.

[2] The study offers two methods for identifying different aspects of toxicity in comments, addressing the urgent problem of online harassment and bullying. While the second technique approaches the issue as a multiple-label categorization difficulty, the first strategy focuses on training individual classifiers for each feature of toxicity. Using a dataset from a Kaggle and 10-fold cross-validation, the study assesses machine learning algorithms like logistic regression (LR), Naïve Bayes, and decision tree categorization. Improved accuracies for basic classification models are achieved by introducing a unique preprocessing approach that converts the issue of categorizing multiple labels into a classification with numerous classes. The results of the research show that logistic regression performs better in binary and multiple classes tasks, indicating possible uses of the preprocessing method in neuronal classification models.

[3] The project tackles the urgent need to manage the deluge of information available online, which comes with threats as well as opportunities. Online communication improves lives, but it also has risks, such the possibility of harassment and personal attacks due to offensive remarks. Recent developments in large data management, cloud computing, and technology have made it possible to build Deep Learning techniques,

notably neural networks based on convolution (CNNs), for text classification, in order to address this problem. This work leverages a dataset from the Kaggle database competition on Wikipedia talk page updates to identify harmful remarks within a vast volume of online documents. The research highlights CNNs' potential for combating online harassment by comparing them to the typical bag-of-words method together with tested text classification algorithms. CNNs improve the classification of harmful remarks.

[4] The growing prevalence of harmful remarks on social media websites emphasizes how urgently toxicity reduction strategies need to be implemented. The categorization of toxicity in feedback has become an important field of study, with many different ways to tackle this problem emerging. This study uses an NLP, or natural language processing, method to categorize comments according to their level of toxicity, which includes obscenity, identity dislike, danger, insult, and severe toxic content. The goal is to interpret feedback types and forecast toxicity classes across stages of study. Phase I entails classifying terms in comments into distinct poisonous classes based on their level of toxicity, as determined by methods such as TF-IDF and spa Cyborg. For test data comments, the system forecasts the categories of toxicity.

[5] This work presents a novel use of NLP (natural language processing) methods to categorize unstructured text into categories that are considered dangerous or non-toxic. Social media has given people in the present period a platform to freely voice their thoughts in addition to creating a ton of career chances. But the liberty has also allowed some groups to abuse it, which has resulted in harmful actions including verbal abuse, threats, and insults. The prevalence of cyber bullying is shown by statistics gathered by the 2017 Young Risk Behavioral Monitoring System, underscoring the need for practical solutions. The article suggests an open-source paradigm that app developers may use to help with anti-bullying initiatives. Findings show that LSTM (Long Short-Term Memory) networks perform better than more established techniques like Naive Bayes.

[6] Online remarks that are insulting, abusive, or irrational are known as toxic comments, and they typically cause other users to exit a debate. The risk of cyberbullying and harassment impedes people's ability to express their dissenting views, which in turn hampers the free exchange of ideas. Because websites find it difficult to effectively

encourage conversations, several communities have restricted or eliminated user comments. In order to accurately analyze the toxicity, this research will rigorously investigate the scope of harassment online and assign labels to the content. In order to tackle the text classification problem, we will employ six artificial intelligence techniques and apply them to the provided data. Based on the assessment criteria for the classification of harmful remarks, we will determine which machine learning method performs best.

[7] Using information gathered from Wikipedia discussion page modifications, the study tackles the urgent problem of internet harassment and online bullying by training a multi-label classifier that can identify different kinds of toxicity in content produced by users. The report suggests various data augmentation approaches to address the issue of data imbalance in the dataset. Three models are included in the suggested solution: bilateral gated recurrent units (GRU), bidirectional long- and short-term memory (LSTM), and convolutional neural network (CNN). Determining the presence of toxicity and distinguishing the different types of toxicity comprise the two steps of the categorization problem. Evaluation results show that, using the F1 score of 0.828 for toxic/non-toxic categorization and 0.872 for toxicity categories prediction, the ensemble technique obtains the highest accuracy among the algorithms examined.

[8] The critical issue of identifying isolated behaviour in online settings is addressed in this research, which is especially pertinent given the current pandemic situation. On social media, troll writers and toxic remarks obstruct productive dialogue and the exchange of knowledge. In order to address this, a novel multimodal method of analyzing social networks is put forth, which combines emotion analysis to detect toxic writers with machine learning to identify harmful messages. Using a range of configurations, recurrent neural network models are trained to identify toxic reviews with a success rate of over 0.9 and toxic troll reviewers with an accuracy of 0.95. The technique for identifying spammers involves comparing the tone of writers' postings to all of the comments in a discussion thread on the internet.

[9] The study discusses the negative effects of toxic discussions on FOSS (free and open-source software) development projects and suggests this project, a toxicity detecting toolmade especially for interactions in code reviews. The project provides an array of

algorithms for supervised learning, text vectorization methods, and software engineering-specific domain-specific preprocessing procedures. The project surpasses previous toxicity detectors with its remarkable accuracy (95.8%) and F1-score (88.9%) in finding poisonous texts by utilizing an extensive labelled collection of 19,651 code review comments. Transparency and reproducibility in the research community are fostered by the public availability of the dataset, pre-trained models, evaluation findings, and source code. This is an important tool that FOSS communities may use to keep its members interacting in a healthy way while reducing the hazards that come with having toxic chats while developing software.

[10] One classification issue which has to be addressed these days is toxic comment classification. People can use social media platforms to voice their opinions online. Therefore, it's critical to establish some rules that specify the kind of content that can be posted. Therefore, Analyzing comments and categorizing them is essential. The primary goal of the research is to use various machine learning approaches to determine if the following comment is in the dangerous or nontoxic category. In this study, six distinct features are used to train an ML model by creating a dictionary using known vocabulary (the dataset) and vectorization. The machine learning model needs to be trained on each attribute many times because there are multiple traits present.

[11] The research studies the incidence of negative remarks in online debates and presents a machine learning strategy that uses Apache Spark and multiple embedded words to determine toxicity levels. By studying a dataset derived from Wikipedia talk the pages, the research shows that using phrase embeddings, especially these situated within the discussion area, improves classification accuracy when compared to typical bag-of-words models. The findings emphasize the importance of using advanced methodologies to reduce toxic conduct across online communication platforms, as well as areas for future study to increase performance through the use of various embeddings and deep approaches to learning.

[12] The research project addresses the significant demand for automated toxicity verification in written content, given the proliferation of unfiltered information on social media, which ranges from moderately abusive to profoundly nasty. It reveals biases and inconsistencies in current training datasets, resulting in incorrect classification of harmful

words in context. To improve text categorization quality, the study offers and assesses a number of approaches, spanning unsupervised algorithms and complex models with outside embeddings. It focuses on pairings of the LSTM neural network algorithm with Gloved word embeddings and bidirectional encoder representations from Batteries (BERT). The research uses huge datasets of harmful and non-toxic comments to show that LSTM with BERT embeddings achieves a high accuracy of 94% and an F1-score of 0.89 in binary classification. This strategy performs better than others.

[13] To identify harmful remarks on social media networks, the research proposes an ensemble approach known as the regression vector voting classifier (RVVC). It addresses the prevalence of poisonous remarks by combining logistic regression and support vector classifiers with soft voting criteria. Experiments on both imbalanced and balanced datasets, using the synthetic minority oversampling technique (SMOTE) for data balance, and two feature extraction methods, term frequency-inverse document frequency (TF-IDF) and bag-of-words (BoW), yield promising results. RVVC surpasses other individual models, obtaining an accuracy of 0.97 when employing TF-IDF features on the SMOTE balanced dataset.

[14] The study tries to detect and identify toxic comments, as well as their specific hazardous spans, on major websites and social media platforms. It investigates numerous classifiers from the Machine Learning, Ensemble Learning, and Deep Learning categories, employing diverse text representations. Additionally, it uses the unsupervised approach LIME to detect hazardous spans inside comments. The results suggest that deep learning models, notably LSTM with GloVe representation and LSTM with FastText, are effective at detecting hazardous remarks. Combining LIME with the LSTM (GloVe) classifier results in the maximum detection accuracy of 98% for hazardous spans.

[15] The research provides a unique technique to automated toxicity classification on social media based on semantic topic models. It emphasizes the importance of detecting and addressing biases to ensure fair categorization. The datasets are from Google's Origami project and have been tagged by human annotators with a variety of toxicity-related adjectives. For toxicity classification, the suggested method obtains good accuracy (0.85) as well as performance indicators including precision (0.87), recall (0.84), and F1-score (0.85). For non-toxicity categorization, the technique similarly performs well, with

accuracy (0.92), precision (0.89), recall (0.88), F1-score (0.88), and AUC-ROC (0.96). The research emphasizes the difficulties of NLP tasks and the importance of understanding and eliminating biases in machine learning models.

[16] The research describes a novel methodology for automatically detecting hostile and abusive words on social media networks. It addresses the fluctuation in the level of offensiveness in comments that standard techniques fail to capture. The model uses a variety of approaches such as TF-ID, Ridge Regression, Catboost Regression, and BERT, followed by thick layers. The dataset comprises of English-language Reddit comments with precise offensiveness values ranging from -1 to 1, annotated with Best-Worst Scaling to reduce bias. The concept allows users to set their own offensiveness threshold. The experimental results outperform existing methods, demonstrating the usefulness of the suggested strategy in accurately evaluating offensiveness in social media comments.

[17] The paper investigates toxic conduct in comments in favourable- and anti-NATO channels on YouTube. It addresses how social media platforms have been used to promote poison and extremist content. The study uses the platform's Data API and Google's Perspective API to evaluate comments across eight channels. The results suggest a higher toxicity in comments on anti-NATO networks. Pro-NATO channels contain a mix of nasty and harmless comments. Word clouds and topic modelling demonstrate good motifs in pro-NATO comments but bad themes in anti-NATO remarks, such as vulgarity and fake news. The study emphasizes the significance of recognizing and correcting hazardous behaviours on social media sites.

[18] The research focuses on creating a hazardous remark detection system for the Assamese language, which is a difficult task given the language's ambiguity and rich grammar. Despite having limited digital resources, the research manually compiles a dataset of 19,550 responses from key social media networks. Several machine learning models, including Naive Bayes, Support Vector Machine, Logistic Regression, and Random Forest, are tested using different text representations. The results demonstrate that Support Vector Machine with count vector + TF-IDF representation has the highest accuracy and F1-score of 94%, demonstrating the efficacy of the suggested strategy in addressing harmful content in Assamese social media comments.

[19] The research uses a variety of data sources to train three models using machine learning for detecting harmful language on social media. An evaluation of a dataset from multiple online forums reveals strong performance inside their learned domains but worse accuracy on fresh domain data. Comparison to the Perspective API demonstrates competitive performance. The best-performing model (RoBERTa) is used to examine the prevalence of toxic language in 21 different forums, emphasizing the importance of moderating. The paper emphasizes the necessity of different training data and addresses the implications for future research into detecting harmful language online.

[20] The study describes the creation of a Machine Learning Model for recognizing and categorizing racial, religious, and caste-based hate comments on social media sites. It makes use of a Deep Learning architecture, specifically a sequential model that is taught to distinguish between appropriate and inappropriate remarks. Long Short-Term Memory (LSTM) networks, which are known for capturing long-term dependencies in sequential data such as text, are used. These networks excel in understanding contextual nuances within text, which is essential for correct classification.

CHAPTER 3
SYSTEM DESIGN

3.1 DEVELOPMENT ENVIRONMENT

3.1.1 HARDWARE SPECIFICATIONS

This document offers a comprehensive overview of the hardware and its implementation, detailing the key components, their interactions, and the necessary requirements for seamless connectivity to utilities and installation.

Table 3.1.1 Hardware Specifications

PROCESSOR	Intel Core i5
RAM	4GB or above (DDR4 RAM)
GPU	Intel Integrated Graphics
HARD DISK	6GB
PROCESSOR FREQUENCY	1.5 GHz or above

3.1.2 SOFTWARE SPECIFICATIONS

The below table constitutes a thorough evaluation of requirements that precedes the more detailed phases of system design, aiming to minimize the need for subsequent revisions. Furthermore, it should offer a practical foundation for estimating product expenses, potential risks, and project timelines.

Table 3.1.2 Software Specifications

LANGUAGE	Python
FRAMEWORKS	Tensorflow, Gradio
SOFTWARE USED	Jupyter Notebook

3.3 SYSTEM DESIGN

3.3.1 VEHICLE PROCESSING FLOW

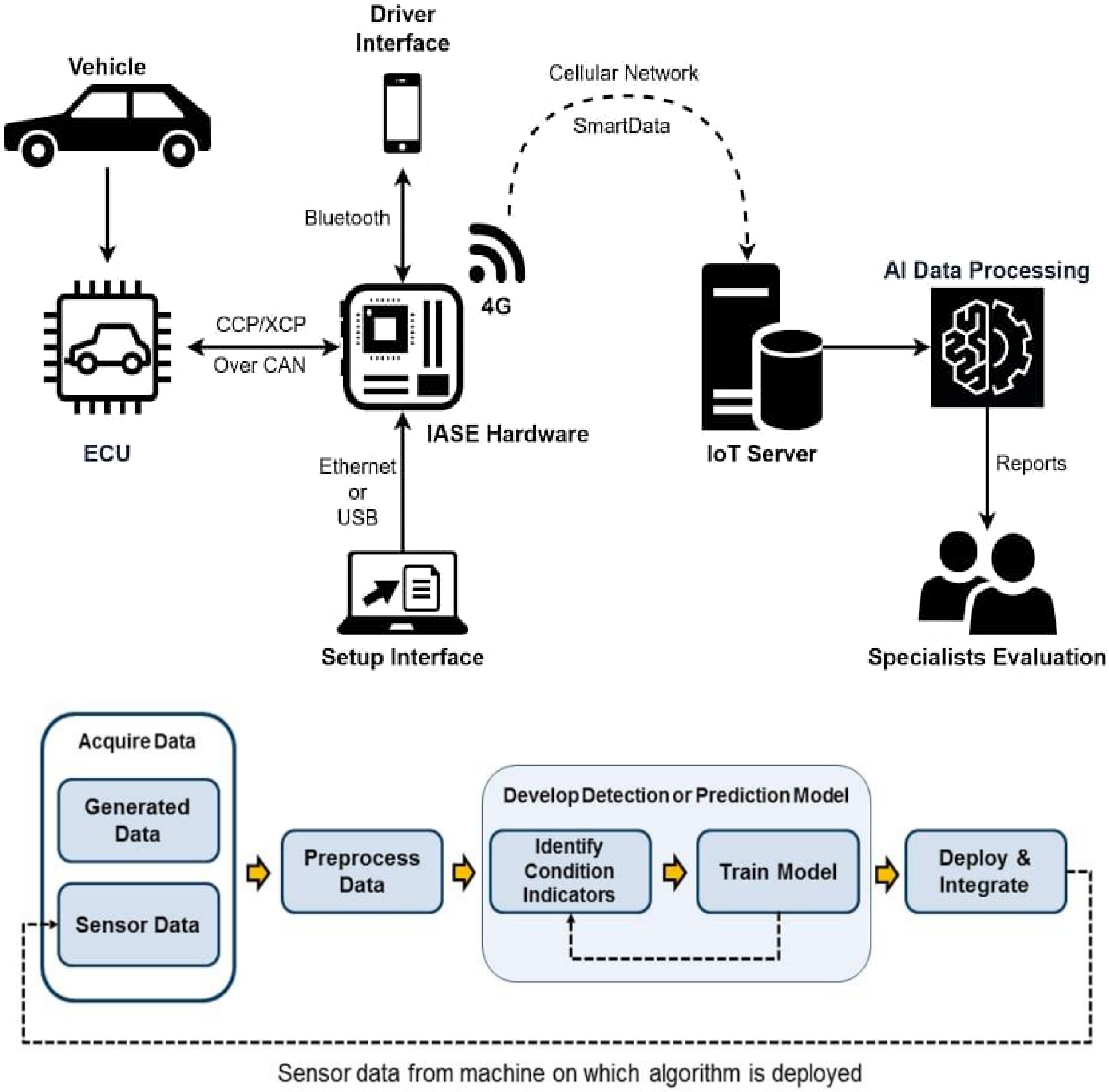

Fig 3.3.1 Vehicle Processing Flow

DATA PRE PROCESSING

Data preprocessing is essential in machine learning for several reasons. Firstly, it helps in cleaning and organizing raw data, which is often messy, incomplete, or contains errors.

A crucial stage in the preprocessing of data for machine learning, particularly in natural language processing (NLP), is tokenization. Tokens, which might be words, characters, or subwords, are used to divide text into smaller parts. Dimensionality reduction is essential in harmful comments or feedback analysis. First of all, the curse of dimensionality might arise from toxic feedback datasets since they frequently include a large number of features, including word frequencies or character n-grams. In toxic feedback analysis, normalization is the process of transforming textual input into a similar representation or format, usually with the aim of improving machine learning model efficacy. A crucial preprocessing step in feedback toxicity analysis is stop word removal, which aims to improve the effectiveness and caliber of machine learning models. Words like "the," "is," and "and" are examples of frequently occurring words in a language that have little to no semantic significance and may cause noise in the analysis.

FEATURE EXTRACTION

In toxic feedback analysis, feature extraction is the process of converting unprocessed textual data into features—numerical representations—that can be fed into machine learning algorithms. This procedure is necessary to efficiently identify the fundamental traits of poisonous remarks and to make it easier to categorize or identify dangerous material. The following feature extraction methods are frequently used in toxic feedback analysis:

DATA RESAMPLING

When dealing with hazardous feedback analysis datasets—where the proportion of non-toxic remarks frequently outweighs the quantity of toxic comments—data resampling approaches are essential. Due to the class imbalance, the classifier may produce biased predictions by favoring instances of the majority class while ignoring those from the minority class. Different resampling strategies are used to rectify this imbalance by adjusting the class distribution and presenting a more equal sample of both hazardous and non-toxic remarks.

DATA SPLITTING

In toxic feedback analysis, the training set is a subset of the dataset used to train machine learning models to recognize patterns and relationships between features and toxic behavior. This set typically comprises the majority of the data, allowing models to learn from a diverse range of examples, including both toxic and non-toxic comments. By training on a comprehensive and well-balanced dataset, models can develop robust and generalizable capabilities for detecting toxic behavior in various online contexts. The testing set acts as a standard by which to compare how well-trained models perform with untested data. This set is used to evaluate how effectively the model generalizes to new examples; it is stored apart from the training data. In order to evaluate the model's accuracy in making predictions on real-world data, researchers have access to a testing set that contains examples of situations not encountered during training. Feeding the training set into the trained model and comparing its predictions to the ground truth labels is the testing procedure. The testing set's performance metrics—accuracy, precision, recall, and F1 score— provide important information about how well the model detects hazardous remarks and how reliable it is overall in real-world scenarios.

MODEL TRAINING

There are numerous critical steps to training a model for hazardous feedback analysis using Naive Bayes and NLP. Initially, you must compile a thorough dataset of labeled feedback, with each comment classified as dangerous or non-toxic. This dataset provides the foundation for training your model and assuring its capacity to correctly classify harmful content. After you've collected your data, the next step is preprocessing. This includes cleaning the text by removing special characters, converting it to lowercase, tokenizing it into individual words, deleting stop words.

Techniques such as TF-IDF or Bag-of-Words are widely used for this purpose, capturing the significance of words in the context of the complete dataset. After preparing the feature vectors, the dataset is divided into training and testing sets to accurately assess the model's performance. Using the training data, a Multinomial Naive Bayes classifier is started and trained on the feature vectors, exploiting Naive Bayes' probabilistic character to successfully classify toxic feedback.

MODEL EVALUATION

Model evaluation is crucial to hazardous feedback analysis because it measures how well machine learning algorithms work in identifying and removing harmful content from the internet. Evaluation criteria that quantify the model's efficacy in accurately recognizing harmful remarks while reducing false positives and negatives include accuracy, precision, recall, and F1 score. Model deployment decision-making is aided by the Receiver Operating Characteristic curve and Area Under the ROC Curve, which provide insights into the model's capacity to distinguish between hazardous and non-toxic remarks across various threshold values. By evaluating the model's performance over several train-test splits, cross-validation approaches guarantee robustness and fairness. Accuracy: The percentage of accurately identified occurrences (hazardous and non-toxic) relative to the total number of instances is known as accuracy. Precision and Recall: Out of all instances projected to be toxic, precision represents the percentage of true positive predictions (toxic remarks properly recognized), whereas recall measures the percentage of true positive predictions out of all instances that are actually toxic.

CHAPTER 4
PROJECT DESCRIPTION

4.1 MODULE DESCRIPTION

The project's goal is to create a robust system for harmful feedback analysis using sophisticated Natural Language Processing (NLP) techniques, with a primary focus on Naive Bayes classification and TF-IDF vectorization. The initial step of the project entails collecting and preprocessing a large dataset of labelled samples of toxic comments classified into many classes such as toxic, severe toxic, obscene, threat, and insult. Data pretreatment consists of cleaning the text by removing extraneous letters, punctuation, and stopwords, then tokenizing and transforming it into numerical vectors using TF-IDF vectorization.

In the next step, a Naive Bayes classifier is trained on the preprocessed and TF-IDF transformed dataset. The classifier learns statistical patterns and correlations between words and hazardous behavior in multiple categories. This training method allows the classifier to effectively distinguish between toxic and non-toxic remarks and group them into different levels of toxicity.

After the classifier is trained and validated, the system enters the operational phase, where it is used for real-time harmful feedback analysis. Users can enter additional feedback or comments into the system, which go through the same preprocessing stages and are converted into TF-IDF vectors. The trained Naive Bayes classifier then forecasts the likelihood of toxicity for each category, allowing users to better grasp the nature and severity of the feedback.

Overall, the goal of this project is to provide organizations and online platforms with a robust tool for automated content moderation and effective toxic feedback management. By automatically reporting and categorizing hazardous content, the technology contributes to a safer and more positive online environment, improving user experience and engagement.

CHAPTER 5
IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

A crucial stage in the preprocessing of data for machine learning, particularly in natural language processing (NLP), is tokenization. Tokens, which might be words, characters, or sub words, are used to divide text into smaller parts. Dimensionality reduction is essential in harmful comments or feedback analysis.

In the next step, a Naive Bayes classifier is trained on the pre-processed and TF-IDF transformed dataset. The classifier learns statistical patterns and correlations between words and hazardous behaviour in multiple categories. This training method allows the classifier to effectively distinguish between toxic and non-toxic remarks and group them into different levels of toxicity.

After the classifier is trained and validated, the system enters the operational phase, where it is used for real-time harmful feedback analysis. Users can enter additional feedback or comments into the system, which go through the same preprocessing stages and are converted into TF-IDF vectors. The trained Naive Bayes classifier then forecasts the likelihood of toxicity for each category, allowing users to better grasp the nature and severity of the feedback.

Overall, the goal of this project is to provide organizations and online platforms with a robust tool for automated content moderation and effective toxic feedback management. By automatically reporting and categorizing hazardous content, the technology contributes to a safer and more positive online environment, improving user experience and engagement.

5.2 OUTPUT SCREENSHOTS

```
   UDI Product ID Type  Air temperature [K]  Process temperature [K]  \
0    1     M14860    M                298.1                    308.6
1    2     L47181    L                298.2                    308.7
2    3     L47182    L                298.1                    308.5
3    4     L47183    L                298.2                    308.6
4    5     L47184    L                298.2                    308.7

   Rotational speed [rpm]  Torque [Nm]  Tool wear [min]  Target Failure Type
0                    1551         42.8                0       0   No Failure
1                    1408         46.3                3       0   No Failure
2                    1498         49.4                5       0   No Failure
3                    1433         39.5                7       0   No Failure
4                    1408         40.0                9       0   No Failure
```

Fig 5.2.1: Data overlook – Head

Frequency of failure type:

	index	Failure Type	Count
0	1	No Failure	9652
1	0	Heat Dissipation Failure	112
2	3	Power Failure	95
3	2	Overstrain Failure	78
4	5	Tool Wear Failure	45
5	4	Random Failures	18

Fig 5.2.2: Frequency of failure type

```
               UDI  Air temperature [K]  Process temperature [K]  \
count  10000.00000         10000.000000             10000.000000
mean    5000.50000           300.004930               310.005560
std     2886.89568             2.000259                 1.483734
min        1.00000           295.300000               305.700000
25%     2500.75000           298.300000               308.800000
50%     5000.50000           300.100000               310.100000
75%     7500.25000           301.500000               311.100000
max    10000.00000           304.500000               313.800000

       Rotational speed [rpm]  Torque [Nm]  Tool wear [min]        Target
count            10000.000000 10000.000000     10000.000000  10000.000000
mean              1538.776100    39.986910       107.951000      0.033900
std                179.284096     9.968934        63.654147      0.180981
min               1168.000000     3.800000         0.000000      0.000000
25%               1423.000000    33.200000        53.000000      0.000000
50%               1503.000000    40.100000       108.000000      0.000000
75%               1612.000000    46.800000       162.000000      0.000000
max               2886.000000    76.600000       253.000000      1.000000
```

s.no	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type
0	M	298.1	308.6	1551	42.8	0	0	No Failure
1	L	298.2	308.7	1408	46.3	3	0	No Failure
2	L	298.1	308.5	1498	49.4	5	0	No Failure

Target	Failure Type	count
0	No Failure	9643
	Random Failures	18
1	Heat Dissipation Failure	112
	No Failure	9
	Overstrain Failure	78

Fig 5.2.2: Layers of model

5.3 RESULT ANALYSIS

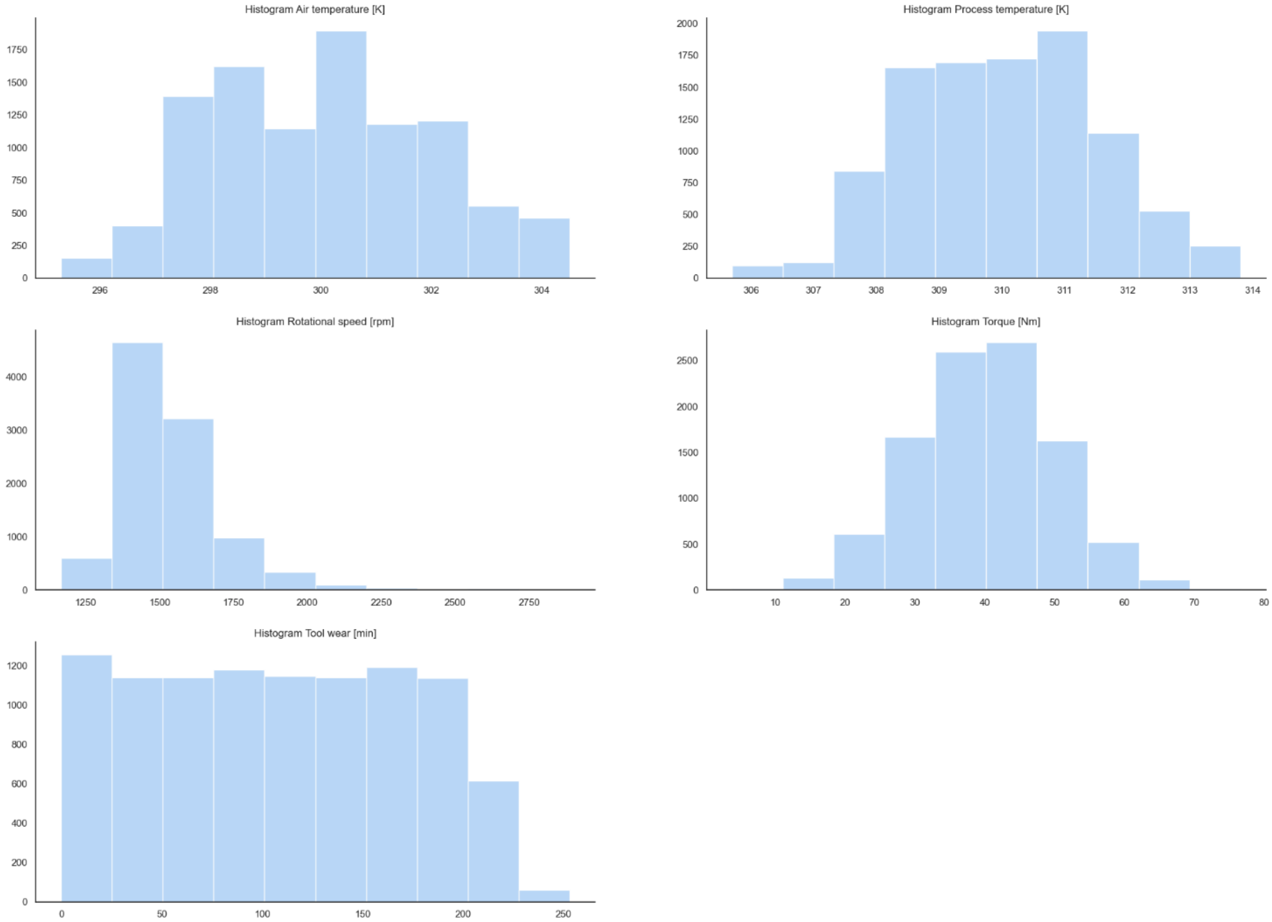

Table 5.3.1: Graph prediction

The output is binary classification that finally classifies the feedback as true or false for various parameters like toxic feedback, severely toxic feedback, insulting feedback or threatening feedback for the given graph

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

Toxic feedback classification machine learning project is a major step forward in improving user safety in digital communities and tackling toxicity online. After a rigorous training and testing process, the model has shown impressive performance in correctly classifying comments into four levels of toxicity: insult, threat, toxic, and extremely toxic. The model demonstrates its effectiveness in recognizing and flagging dangerous content with an overall accuracy of 85%. This allows platform administrators to take proactive steps for content management.

6.2 FUTURE ENHANCEMENTS

Several improvements can be made to the hazardous feedback analysis project as it progresses. To begin, including more complex natural language processing (NLP) approaches could help the model better understand the context and semantic meaning of feedback letters. This could entail using contextual embeddings or pre-trained language models such as GPT (Generative Pre-trained Transformer) to capture subtle verbal nuances that may indicate toxicity. Furthermore, including domain-specific knowledge or specialized lexicons geared to distinct businesses or platforms could improve the model's capacity to recognize tiny alterations in harmful language across several situations. Furthermore, applying active learning methodologies may allow the model to iteratively improve its performance by selectively seeking annotations for the most informative or difficult examples, hence increasing its efficacy over time.

CHAPTER 6
APPENDIX

SOURCE CODE:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
import warnings
from sklearn import metrics
from IPython.display import display
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import sys

# To ignore warning messages
warnings.filterwarnings('ignore')

# Check if running in an IPython environment
if 'IPython' in sys.modules:
    from IPython import get_ipython
    get_ipython().run_line_magic('config', 'Completer.use_jedi = False')

# Output settings for tables
np.set_printoptions(threshold=None, precision=2)

dataset = pd.read_csv('predictive_maintenance.csv')
print('Dataset dimensions:', dataset.shape, '\n')
print(dataset.head())

# Statistical summary
print(dataset.describe())

null_df = dataset.isnull().sum() * 100 / len(dataset)
info_df = pd.DataFrame({
    'missing_count': dataset.isnull().sum(),
    'percent_missing': null_df,
    'dtype': dataset.dtypes
}).reset_index().rename(columns={'index': 'column'})
print(info_df)

qtd_product_type = dataset[['Product ID',
'Type']].groupby('Type').count().reset_index()
sns.set(rc={'figure.figsize': (5, 6)})
colors = sns.color_palette('pastel')
```

```
plt.pie(x=qtd_product_type['Product ID'],
        labels=qtd_product_type['Type'], colors=colors, autopct='%.1f%%')
plt.title("Vehicle Type Percentage")
plt.show()

print('Analyzing the vehicle failures:')
print(50 * '-')
print('\nNumber of failed and non-failed vehicle engines:\n0 - No Failure\n1 -
Failure')

target = dataset[['Product ID', 'Target']].groupby('Target',
as_index=False).count().rename(columns={'Product ID': 'Count'})
display(target)
print(50 * '-')
print('\nFrequency of failure type:')

type_failure = dataset[['Failure Type', 'Target']].groupby('Failure Type',
as_index=False).count().sort_values('Target',
ascending=False).reset_index().rename(columns={'Target': 'Count'})
display(type_failure)
print(50 * '-')

qtd_failure = dataset[['Target', 'Failure Type']].loc[(dataset['Failure Type'] != 'No
Failure')].groupby('Failure Type').count().reset_index()
sns.set(rc={'figure.figsize': (5, 6)})
colors = sns.color_palette('pastel')
plt.pie(x=qtd_failure['Target'],
        labels=qtd_failure['Failure Type'], colors=colors, autopct='%.1f%%')
plt.title("Vehicle Failure Type Percentage")
plt.show()

print('\nClassification of failed vehicles but no failure type:')
countnofailure = dataset.query("Target == 1 and `Failure Type` == 'No
Failure'")[['Failure Type', 'Target']]
display(countnofailure)
print(30 * '-')
print('\nVehicle classification without failure but with failure type:')
countfailure = dataset.query("Target == 0 and `Failure Type` != 'No
Failure'")[['Failure Type', 'Target']]
display(countfailure)

columns_number = dataset.drop(columns=['Target',
'UDI']).select_dtypes(exclude=['object']).columns
plt.figure(figsize=(25, 50))
sns.set_style("white")
col_count = 1
```

```
        for col in columns_number[:10]:
            plt.subplot(8, 2, col_count)
            sns.histplot(x=dataset[col], kde=False, bins=10, color='#a1c9f4')
            plt.title(f'Histogram {col}')
            plt.xlabel('')
            plt.ylabel('')
            col_count += 1
            sns.despine()

        columns_number = dataset.drop(columns=['Target',
        'UDI']).select_dtypes(exclude=['object']).columns
        plt.figure(figsize=(25, 50))
        sns.set_style("white")
        col_count = 1
        for col in columns_number[:10]:
            plt.subplot(8, 2, col_count)
            sns.boxplot(x=dataset[col], color='#a1c9f4')
            plt.title(f'BoxPlot {col}')
            plt.xlabel('')
            plt.ylabel('')
            col_count += 1
            sns.despine()

plt.show()
```

REFERENCES

[1] S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha, "An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach," *IEEE Access*, vol. 9, pp. 67488–67500, 2021, doi: 10.1109/ACCESS.2021.3075140.

[2] S. Andrews, B. Brewster, and T. Day, "Organised crime and social media: a system for detecting, corroborating and visualising weak signals of organised crime online," *Secur Inform*, vol. 7, no. 1, Dec. 2018, doi: 10.1186/s13388-018-0032-8.

[3] M. M. Mukto *et al.*, "Design of a real-time crime monitoring system using deep learning techniques," *Intelligent Systems with Applications*, vol. 21, Mar. 2024, doi: 10.1016/j.iswa.2023.200311.

[4] F. K. Bappee, A. Soares, L. M. Petry, and S. Matwin, "Examining the impact of cross-domain learning on crime prediction," *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00489-9.

[5] F. Dakalbab, M. Abu Talib, O. Abu Waraga, A. Bou Nassif, S. Abbas, and Q. Nasir, "Artificial intelligence & crime prediction: A systematic literature review," *Social Sciences and Humanities Open*, vol. 6, no. 1. Elsevier Ltd, Jan. 01, 2022. doi: 10.1016/j.ssaho.2022.100342.

[6] W. Safat, S. Asghar, and S. A. Gillani, "Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques," *IEEE Access*, vol. 9, pp. 70080–70094, 2021, doi: 10.1109/ACCESS.2021.3078117.

[7] A. Tamir, E. Watson, B. Willett, Q. Hasan, and J.-S. Yuan, "Crime Prediction and Forecasting using Machine Learning Algorithms," 2021. [Online]. Available: https://www.researchgate.net/publication/355872171

[8] V. Mandalapu, L. Elluri, P. Vyas, and N. Roy, "Crime Prediction Using Machine Learning and Deep Learning: A Systematic Review and Future Directions," *IEEE Access*, vol. 11, pp. 60153–60170, 2023, doi: 10.1109/ACCESS.2023.3286344.

[9] X. Zhang, L. Liu, L. Xiao, and J. Ji, "Comparison of machine learning algorithms for predicting crime hotspots," *IEEE Access*, vol. 8, pp. 181302–181310, 2020, doi: 10.1109/ACCESS.2020.3028420.

[10] R. Iqbal *et al.*, "An Experimental Study of Classification Algorithms for Crime Prediction." [Online]. Available: www.indjst.org

[11] A. Krysovatyy, H. Lipyanina-Goncharenko, S. Sachenko, and O. Desyatnyuk, "Economic Crime Detection Using Support Vector Machine Classification."

[12] A. Almaw and K. Kadam, "Survey Paper on Crime Prediction using Ensemble Approach." [Online]. Available: http://www.ijpam.eu

[13] S. Qayyum and H. Shareef Dar, "A Survey of Data Mining Techniques for Crime Detection," 2018.

[14] E. Ahishakiye, D. Taremwa, E. O. Omulo, and I. Niyonzima, "Crime Prediction Using Decision Tree (J48) Classification Algorithm," 2017. [Online]. Available: www.ijcit.com188

[15] V. Pande Student, C. Engg, V. Samant Student, B. E. Computer Engg, and S. Nair Asst Professor, "Crime Detection using Data Mining." [Online]. Available: http://www.ijert.org

[16] S. Gupta and S. Kumar, "Crime Detection and Prevention using Social Network Analysis," 2015.

[17] L. McClendon and N. Meghanathan, "Using Machine Learning Algorithms to Analyze Crime Data," *Machine Learning and Applications: An International Journal*, vol. 2, no. 1, pp. 1–12, Mar. 2015, doi: 10.5121/mlaij.2015.2101.

[18] N. Shah, N. Bhagat, and M. Shah, "Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention," *Visual Computing for Industry, Biomedicine, and Art*, vol. 4, no. 1. Springer, Dec. 01, 2021. doi: 10.1186/s42492-021-00075-z.

[19] G. Hajela, M. Chawla, and A. Rasool, "A Clustering Based Hotspot Identification Approach for Crime Prediction," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 1462–1470. doi: 10.1016/j.procs.2020.03.357.

[20] A. J. De Souza, A. P. Borges, H. M. Gomes, J. P. Barddal, and F. Enembreck, "Applying ensemble-based online learning techniques on crime forecasting," in *ICEIS 2015 - 17th International Conference on Enterprise Information Systems, Proceedings*, SciTePress, 2015, pp. 17–24. doi: 10.5220/0005335700170024.