International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-12 Issue-2, July 2023

# Product Review Classification using Machine Learning and Statistical Data Analysis

**Kajal Singh**

***Abstract:*** ***The aim of the paper is to implement and analyze the machine learning models for product review dataset. The project focuses on binary classification, multi-class classification, and clustering approaches to analyze and categorize product reviews. The performance of the models over each of the five classification tasks is measured by the 5-fold cross-validation scores over the training data.***

***Keywords: Machine Learning, Classification, Clustering, Product***

## I. INTRODUCTION

Machine learning has revolutionized the field of data analysis and has proven to be an effective tool for solving complex classification and clustering problems. In recent years, there has been a significant increase in the availability of datasets, which has led to an explosion of research in the field of dataset classification using machine learning models. In this review paper, we aim to provide an overview of the most commonly used machine learning models for Binary, Multiclass and Clustering dataset classification, their strengths, weaknesses, and limitations based on the models' performance. We will also discuss various preprocessing techniques that can be applied to improve the performance of these models.

## II. RELATED WORK

[1] Machine learning has become an increasingly popular technique for solving classification problems. In classification problems, the goal is to predict a categorical label or class for a given input. [3] For products review dataset, we have implemented the models to predict the overall rating based on the cutoff values and have clustered the dataset and measured model's accuracy in terms of Silhouette score. [5] The classification problems are ubiquitous in many fields, such as finance, healthcare, and e-commerce.

### 2.1 Binary Classification

[4] Binary classification is a type of classification problem in which the goal is to predict one of two possible outcomes, such as 1 or 0, yes or no, true or false, or positive or negative.

**Manuscript received on 08 March 2023 | Revised Manuscript received on 21 June 2023 | Manuscript Accepted on 15 July 2023 | Manuscript published on 30 July 2023.**
*Correspondence Author(s)
**Kajal Singh*,** Dartmouth College Hanover, NH, USA. E-mail: Kajal.singh.th@dartmouth.edu, ORCID ID: 0009-0001-2274-8664



*Retrieval Number: 100.1/ijrte.A75300512123*
*DOI: 10.35940/ijrte.A7530.0712223*
*Journal Website: www.ijrte.org*

IoT = Services + Data + Networks + Sensors

Binary classification is used in many fields, such as fraud detection, spam filtering, and medical diagnosis. [6] There are several machine learning algorithms that can be used for binary classification, including logistic regression, support vector machines, random forest, decision trees, and neural networks.

[3] In this paper, we have implemented binary classification for 4 different cutoff values: 1,2,3,4 to predict the overall ratings: 0 or 1. The cutoff is not an input to the model, but to the experiment. For example, when cutoff=3, all samples with a rating <= 3 will have label 0, and all samples with a rating > 3 have label 1.[5] The model performance has been reported of at least three different classifiers for each of the four cutoffs. For each classifier, we have reported the confusion matrix, ROC, AUC, macro F1 score, and accuracy for the best combination of hyperparameters using 5-fold cross-validation.

### 2.2 Multiclass classification

Multiclass classification is a type of classification problem in which the goal is to predict one of the several possible outcomes. For example, in a medical diagnosis problem, a patient's condition may be classified as normal, mild, moderate, or severe. Multiclass classification is used in many fields, such as image recognition, speech
recognition, and natural language processing. There are several machine learning algorithms that can be used for multiclass classification, including k-nearest neighbors, decision trees, random forests, and neural networks.

[6] For our project, we have turned the binary classifier into a multiclass classifier where the target classes are 1,2,3,4,5. The product rating has been classified on a five-class scale. For each classifier, we have reported the confusion matrix, ROC, AUC, macro F1 score, and accuracy for the best combination of hyperparameters using 5-fold cross-validation. For the multiclass classification task, we will also show 6 curves in one plot: 5 curves from each category and the average curve.

### 2.3 Clustering

[6] Clustering is a type of unsupervised learning in which the goal is to group similar data points together. Clustering is often used in exploratory data analysis, customer segmentation, and anomaly detection. There are several clustering algorithms that can be used, namely k-means clustering, hierarchical clustering, and density-based clustering. In our task, we will cluster the product reviews in the test dataset. We will need to create word features from the data and use that for k-means clustering. Clustering will be done by product types, i.e., in this case, the labels will be product categories.

*Published By:*
*Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)*

The Silhouette score and Rand index will be used to analyze the quality of clustering. In this review paper, we will explore the different machine learning algorithms that can be used for binary classification, multiclass classification, and clustering. We will examine the strengths and weaknesses of each algorithm and provide guidance on when to use each one. We will also discuss the various evaluation metrics that can be used to assess the performance of these algorithms. Finally, we will state the results, analysis and conclusions.

## III. METHODS

[2] Machine learning has become a powerful tool in data analysis, providing efficient solutions to various classification problems. In this section, we focus on three popular classification models: logistic regression, decision tree, and random forest. We compare their performance on amazon real world dataset and highlight their strengths and limitations.

### 3.1 Logistic Regression

Logistic regression is a widely used classification model that predicts the probability of a binary outcome

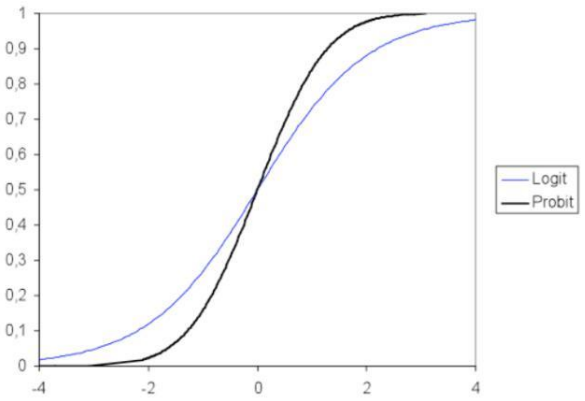


**Figure 1. shows the difference for a logit and a probit model for different values [-4,4]. Both models are commonly used in logistic regression.**

based on one or more independent variables. It is a linear model that uses a logistic function to convert the output into a probability score. Logistic regression is computationally efficient and easy to interpret, making it a popular choice for many classification tasks. However, it assumes a linear relationship between the independent variables and the log odds of the outcome. In our project, we have used logistic regression for Binary and Multiclass classification using Grid search CV for hyperparameter tuning. For the Models, we have chosen Hypermeters as [0.1, 1.0, 10.0] and trained our models for choosing the best hyperparameter.

### 3.2 Decision Tree

A decision tree is a hierarchical model that partitions the dataset into smaller subsets based on the values of the independent variables. It constructs a tree-like structure of decisions and outcomes, where each internal node represents a decision based on a specific feature, and each leaf node represents a classification outcome. Decision trees are easy to interpret and can handle both binary and multi-class classification problems. However, they are prone to overfitting, especially when the tree is deep, and the dataset is complex. In our project, we have implemented decision tree for the Binary and Multiclass classification and evaluated the classifier using 5-fold cross validation. We will choose Hyperparameter as the max Depth for [10, 50, 100] values.

### 3.3 Random Forest

[4] Random Forest is a learning method that combines multiple decision trees to improve the classification performance. It generates a set of decision trees by randomly sampling the training data and the features at each and every split. The final classification result is obtained by averaging the predictions of all the trees in the forest. Random forest is less prone to overfitting than decision trees and it can also handle large datasets with high dimensionality. However, it might be computationally expensive and difficult to interpret sometimes.

In this review paper, we compared the performance of logistic regression, decision tree, and random forest on our datasets. Logistic regression is a simple and efficient model that works well for binary classification tasks, while decision tree and random forest can handle complex datasets with high dimensionality. Decision trees are easy to interpret, but highly prone to overfitting, while random forest is somewhat lesser prone to overfitting but can be computationally expensive. We will choose the right classification model depending on the nature of the dataset, specific classification task and models performance.

### 3.4 Performance Metrics

The performance of the predictive models can be measured via different metrics such as Accuracy, Macro F-score, AUC_ROC Score and Confusion Matrix, Silhouette score and Rand index.

#### *3.4.1 Accuracy*

Accuracy is a metric that is used in machine learning to measure the performance of a classification model. It is defined as the proportion of rightly classified instances out of the total number of instances occurred. [3] For binary classification, when there are two possible outcomes (yes or no/ 0 or 1), the accuracy is calculated as: Accuracy = (true positives + true negatives) / (true positives + false positives + true negatives + false negatives) where true positives (or TP) are the number of correctly classified positive instances, false positives (or FP) are the number of negative instances that were incorrectly classified as positive, true negatives (or TN) are the number of correctly classified negative instances, and false negatives (or FN) are the number of positive instances that were incorrectly classified as negative. For multiclass classification, where there are more than two possible classes (In our project case, there are 5 classes) the accuracy will be calculated in different ways depending on the problem and the goals of the analysis. Here are two commonly used methods:

#### *3.4.2 Macro-averaged accuracy*

This measures the accuracy for each class separately and then averages them equally. This is useful when you want to ensure that each class is equally important and that the model performs well on all of these classes.

*Retrieval Number: 100.1/ijrte.A75300512123*
*DOI: 10.35940/ijrte.A7530.0712223*
*Journal Website: www.ijrte.org*



*Published By:*
*Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)*

The formula for macro-averaged accuracy is shown below:
Average Macro Accuracy = sum_i^C (TP_i) / sum_i^C (TP_i + FP_i)
where C = total number of classes
TP_i = number of true positives for class i,
FP_i = number of false positives for class i.

### 3.4.3 *Weighted accuracy*

[5] This measures the accuracy for each class separately and then averages them using the class frequencies as weights. This is useful when you want to consider the imbalance in dataset and give weight to the classes with more crucial instances. The formula for weighted accuracy is shown below:
Weighted Accuracy = sum_i^C (w_i * TP_i) / sum_i^C (w_i * (TP_i + FP_i))
Where w_i = weight of class i,
C = total number of classes
TP_i = number of true positives for class i,
FP_i = number of false positives for class i.

### 3.4.4 *Macro F1-Score*

The F1-score is a commonly used evaluation metric that combines precision and recall into a single score. The [5] Macro F1-score is a variant of the F-score used in multiclass classification datasets. It calculates the F-score for each class separately and then takes the average of the scores to get an overall performance measure.
[3] For binary classification, the F-score is defined as the harmonic mean of precision and recall, and it is calculated as shown below:
F-score = 2 * (precision * recall) / (precision + recall)
where precision = proportion of true positive predictions out of all positive predictions,
and recall = proportion of true positive predictions out of all actual positive instances.
For multiclass classification, we can calculate the Macro F-score as shown below:
We can calculate precision, recall, and F-score for each class i:
precision_i = TP_i / (TP_i + FP_i)
recall_i = TP_i / (TP_i + FN_i) F-score_i = 2 * (precision_i * recall_i) / (precision_i + recall_i)
where TP_i, FP_i, and FN_i are the true positives, false positives, and false negatives, respectively.
Macro F-score is further calculated as the average of the F-scores across all classes:
Macro F-score = (F-score_1 + F-score_2 + ... + F-score_C) / C
where C = total number of classes.
The Macro F1-score is a useful metric when we want to ensure that the model performs well on all classes, regardless of imbalance or size of the classes.

### 3.4.5 *AUC ROC Score*

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is an evaluation metric in binary and multiclass classification problems which measures the quality of the model's predictions across different probability and thresholds. The AUC-ROC score ranges from 0 to 1, in which values closer to 1 indicate better performance.

[4] In binary classification, the ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) for different probability thresholds.
TPR: proportion of true positives that are correctly identified
FPR = proportion of false positives that are incorrectly identified.
The AUC-ROC score is then calculated as the area under this curve.
In multiclass classification, we can calculate the AUC-ROC score in different ways depending on the problem and the goals of the analysis. One common approach is to use the one-vs-all method, in which we treat each class as positive and the rest as negative. Further, calculate the AUC-ROC score for each class separately. The final AUC-ROC score is finally calculated as the average of the scores across all classes. It is useful when the classes are imbalanced or the costs of false positives and false negatives are different, because it allows us to compare the performance of different models and choose the one that best balances these trade-offs. However, it may not be suitable for all problems, especially when the class distributions are more skewed or the decision threshold is not important.

### 3.4.6 *Confusion Matrix*

A confusion matrix is a table that summarizes the performance of a binary or multiclass classification model by comparing the predicted labels with the true labels. It is a useful tool for evaluating the model's accuracy, precision, recall, and F1-score for each class and identifying the types of errors that the model has made.
In a binary classification problem, a confusion matrix has four possible outcomes:
True Positive (TP): the model correctly predicted the positive class
False Positive (FP): the model predicted the positive class but the true label was negative
True Negative (TN): the model correctly predicted the negative class
False Negative (FN): the model predicted the negative class but the true label was positive
The confusion matrix for binary classification can be represented as follows:

**Table 1. A Layout of Confusion Matrix**

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positive (TP) | False Positive (FP) |
| Predicted Negative | False Negative (FN) | True Negative (TN) |

In our multiclass classification, the confusion matrix has 5 possible outcomes, depending on the number of classes. The diagonal elements of the matrix represent the number of correctly predicted instances for each class, while the off-diagonal elements represent the number of incorrectly predicted instances.



Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The confusion matrix for multiclass classification is shown below:

| | Actual Class 1 | Actual Class 2 | ... | Actual Class N |
|---|---|---|---|---|
| Predicted Class 1 | True Positive (TP1) | False Positive (FP1) | ... | False Positive (FPN) |
| Predicted Class 2 | False Positive (FP2) | True Positive (TP2) | ... | False Positive (FPN) |
| ... | ... | ... | ... | ... |
| Predicted Class N | False Positive (FPN) | False Positive (FPN) | ... | True Positive (TPN) |

The elements of the confusion matrix can be used to calculate various performance metrics, namely accuracy, precision, recall, and F1-score for each class, as well as overall metrics such as macro-averaged and micro-averaged scores. The confusion matrix is an important tool for visualizing and interpreting the performance of the model, as well as identifying areas for improvement in the algorithm.

*3.4.7 Silhouette Score*

The silhouette score is a metric used to evaluate the quality of clusters produced by clustering algorithms. It measures how well each data point is clustered relative to the other points in its cluster as compared to the points in the other clusters. The silhouette score ranges from -1 to 1, where a score of 1 indicates that the data point is well-clustered and a score of -1 indicates that it is badly-clustered. The silhouette score for a single data point is calculated as shown below: Calculate the mean distance between the data point and all other points in its cluster. This is called the "intra-cluster distance" and denoted by a(i).

Calculate the average distance for all the clusters. This is called the "inter-cluster distance" and denoted by b(i).

Calculate the silhouette score for the data points:

(b(i) - a(i)) / max(a(i), b(i)).

The overall silhouette score for a set of data points is calculated as the mean of the silhouette scores for each point. A high silhouette score indicates that the data points are well-clustered, with points in the same cluster being close to each other and far from points in other clusters.

The silhouette score can be important metric for comparing the quality of different clustering algorithms and for choosing the optimal number of clusters. A high silhouette score suggests that the clusters are well-separated and that the data points are assigned to the correct clusters. Conversely, a low silhouette score shows that the clusters are overlapping or poorly separated, and that the data points are not assigned to the correct clusters. However, the silhouette score has a few limitations, such as being sensitive to the choice of distance metric and the type of clustering algorithm used.

*3.4.8 Rand Index*

The Rand index is used to evaluate the similarity between two sets of cluster assignments, majorly the predicted cluster assignments produced by a clustering algorithm and the true cluster assignments. This index calculates the proportion of pairs of data points that are either correctly assigned to the same cluster or correctly assigned to different clusters by both the true and predicted cluster assignments.

The Rand index ranges from 0 to 1, where a score closer to 1 indicates that the predicted cluster assignments are identical to the true cluster assignments while score closer to 0 indicates that the predicted cluster assignments are completely random.

The Rand index is calculated as follows:

Let n be the total number of data points.

Let a be the number of pairs of data points that are in the same cluster in both the true and predicted cluster assignments.

Let b be the number of pairs of data points that are in different clusters in both the true and predicted cluster assignments.

Rand index = (a+b)/n

where n is the total number of pairs of data points.

The Rand index is an intuitive metric for evaluating clustering algorithms, but it has some limitations, such as being sensitive to the number of clusters and the distribution of data points within clusters. Additionally, it can be affected when the number of clusters is large or the data points are poorly separated. Therefore, the Rand index should be used in addition with other clustering evaluation metrics, such as the silhouette score and the adjusted mutual information score.

## IV. RESULTS

The product reviews have been classified and clustered into different classes using Binary, Multiclass Classification and clustering. After running the predictive models, the performance has been measured via different metrics such as Accuracy, Macro F-Score, and AUC_ROC score, Silhouette Score and Rand Index.

### A. Results and analysis

*a. Binary Classification*

Choosing the best hyperparameter for Binary Classification algorithm is crucial for achieving good performance on unseen data. We are using Grid Search CV approach for selecting the best hyperparameters from a set of candidate values. In our code, we have defined a set of candidate values for each hyperparameter based on the type of algorithm being used. For example, for Logistic Regression, we are testing three different values of the C parameter. Similarly, for Decision Tree, we are testing three different values of the max_depth parameter, and for Random Forest, we are testing three different values of the n_estimators parameter. The best hyperparameter is selected based on the performance of the algorithm on a validation set. Grid Search CV function will search through all the possible combinations of hyperparameters and evaluate the algorithm on each combination using 5 fold cross-validation. It will then select the combination of hyperparameters that gives the best performance on the validation set.

*Retrieval Number: 100.1/ijrte.A75300512123*
*DOI: 10.35940/ijrte.A7530.0712223*
*Journal Website: www.ijrte.org*



*Published By:*
*Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)*

We have used these hyperparameters to train the final model on the entire training set and evaluate its performance on the test set. After experimentation, we have seen that the choice of hyperparameters can have a significant impact on the performance of the algorithm. Therefore, it's a good practice to experiment with different sets of candidate values for each hyperparameter and choose the one that gives the best performance out of all.

For this paper, we have observed that logistic regression classifier gives the best performance for all 4 cutoffs among the 3 implemented classifiers.

i. Best hyperparameters: {'C': 0.1}: This indicates the best hyperparameters found by the Grid Search CV algorithm. In this case, the best value of the regularization parameter C was found to be 0.1 for all the classes.
ii. ROC AUC score: This is the Receiver Operating Characteristic (ROC) curve's area under the curve (AUC) score, which measures the model's ability to distinguish between positive and negative samples. The score ranges from 0 to 1, with a score of 0.5 indicating a random prediction and a score of 1 indicating perfect prediction. In our case, the score is above 0,5 in all cases, hence, it can be considered moderately good score.
iii. Confusion matrix: This summarizes the model's predictions on the test set. The confusion matrix shows the number of true positives, false positives, true negatives, and false negatives. For example, in the case of Binary classification with cutoff<=1, the confusion matrix shows that the model correctly predicted 2529 samples as true positives and 17693 samples as true negatives. It also incorrectly predicted 2195 samples as false positives and 934 samples as false negatives.
iv. Macro F1 score: This is the harmonic mean of precision and recall that computes the average F1 score across all classes, with each class weighted equally. The F1 score ranges from 0 to 1, with a score of 1 indicating perfect precision and recall. For our models, the macro F1 Score is considered to be good as they are more than 0.7.
v. Accuracy: This is the overall accuracy of the model on the test set measuring the proportion of correct predictions among all samples. The accuracy ranges from 0 to 1, with a score of 1 indicating perfect accuracy. In our second cutoff case of binary classification case, the accuracy is 0.82, which is a good score. However, it's important to note that accuracy can be misleading if the classes are imbalanced or if the cost of false positives and false negatives is different. Therefore, it's always important to look at other evaluation metrics like precision, recall, F1 score, and ROC AUC score in addition to accuracy.

### B. Multiclass Classification

The prediction models' output reports the performance of three different classifiers, namely logistic regression, decision tree, and random forest for multiclass classification. Logistic Regression model: F1 score is reported as 0.56, which is a measure of the model's accuracy. The hyperparameter that produced the best results is C=0.1, and the macro F1 score and accuracy of the model are also reported as 0.56 and 0.57, respectively. Confusion matrix: It gives detailed information about the performance of the model, showing the number of instances that belong to each true class and the number of instances that were classified into each predicted class. For example, the element in the first row and second column (229) indicates that 229 instances that belong to the first class were misclassified as belonging to the second class by the logistic regression model.

*b. Decision tree model:*

F1 score is reported as 0.44, and the best hyperparameters are {'max_depth': 10}, which is the maximum depth of the tree. The macro F1 score and accuracy of the model are also reported as 0.45 and 0.45, respectively.

Confusion matrix: It gives information about the performance of the model, indicating how many instances were classified into each predicted class.

Random Forest Model:

F1 score is reported as 0.54, and the best hyperparameters are {'n_estimators': 200}, which is the number of decision trees used in the ensemble. The macro F1 score and accuracy of the model are also reported as 0.54 and 0.55, respectively. [5] Confusion matrix: It gives information about the performance of the model, indicating how many instances were classified into each predicted class. In summary, the output provides a comprehensive evaluation of the performance of three different classifiers on a multiclass classification problem, including their F1 score, hyperparameters, macro F1 score, accuracy, and confusion matrix.

### C. Clustering

[6] The clustering model output is related to the evaluation of a clustering algorithm, and two metrics are reported: Silhouette score and Rand index. Silhouette score: It is a measure of how well the instances are clustered, where a higher score indicates better clustering. The value of Silhouette score reported is 0.64, which is a relatively good score, indicating that the instances are well-clustered. Rand index: It is another measure of how well the instances are clustered, where a higher value indicates better clustering. However, the reported value of Rand index is quite low, approximately 1.97e-07, indicating that the clustering algorithm may not have performed well in this specific case. In summary, the output suggests that the clustering algorithm has produced a reasonably good clustering of the instances based on the Silhouette score, but the Rand index suggests that there is still room for improvement. It is important to note that both Silhouette score and Rand index should be used together to provide a comprehensive evaluation of the clustering algorithm's performance.

## V. CONCLUSION

In the product review classification and clustering project, we evaluated the performance of binary classification, multiclass classification, and clustering algorithms on amazon's dataset.

For binary classification, we used logistic regression, decision tree, and random forest algorithms. Among these, logistic regression performed the best with a ROC AUC score of 0.74, macro F1 score of 0.77, and accuracy of 0.87. The decision tree and random forest algorithms also Performed good, but with lower scores than logistic regression. Hence, we can choose logistic regression for binary classification of the dataset. For multiclass classification, we used logistic regression, decision tree, and random forest algorithms. Once again, logistic regression performed the best with a macro F1 score of 0.56 and accuracy of 0.57. The decision tree and random forest algorithms had comparatively lower scores than logistic regression. [6] For clustering, we calculated silhouette score and Rand index to evaluate the performance of the algorithm. The silhouette score was 0.64, which indicates that the clusters are well-separated. However, the Rand index was very low. Overall, logistic regression performed the best among the binary and multiclass classification.

## DECLARATION

| | |
|---|---|
| Funding/ Grants/ Financial Support | No, I did not receive. |
| Conflicts of Interest/ Competing Interests | No conflicts of interest to the best of our knowledge. |
| Ethical Approval and Consent to Participate | No, the article does not require ethical approval and consent to participate with evidence. |
| Availability of Data and Material/ Data Access Statement | Not relevant. |
| Authors Contributions | I am only the sole author of the article. |

## REFERENCES


1. https://www.researchgate.net/publication/342890321_Machine_Learning_A_Review_of_Learning_Types
2. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
3. https://machinelearningmastery.com/types-of-classification-in-machine-learning/
4. https://en.wikipedia.org/wiki/Binary_classification
5. https://scikitlearn.org/stable/auto_examples/model_selection/plot_roc.html
6. https://www.educative.io/answers/classification-vs-clustering


## AUTHOR PROFILE

**Kajal Singh** is a highly motivated graduate student pursuing a Masters degree in Engineering Management from Dartmouth College. With a strong background in Electronics and Communication Engineering, Kajal brings a diverse set of skills to her current academic pursuits. Prior to her academic journey, Kajal gained valuable experience as a Project Manager and Senior Business Analyst at Schneider Electric. Her three years of experience in the field allowed her to gain expertise in areas such as data analytics, and product management. In addition to her academic pursuits, Kajal is a dedicated fitness enthusiast with a passion for sports cars. Kajal is eager to continue exploring her passion for AI and ML and is confident that her academic background and practical experience will help her make a meaningful impact in the field.