# Project Report

**Project Name:** Manufacturing Defect Detection

**Team Members:** Dedeepya Vesangi, Harshitha Manaswini vadavalli, Nithin Aleti

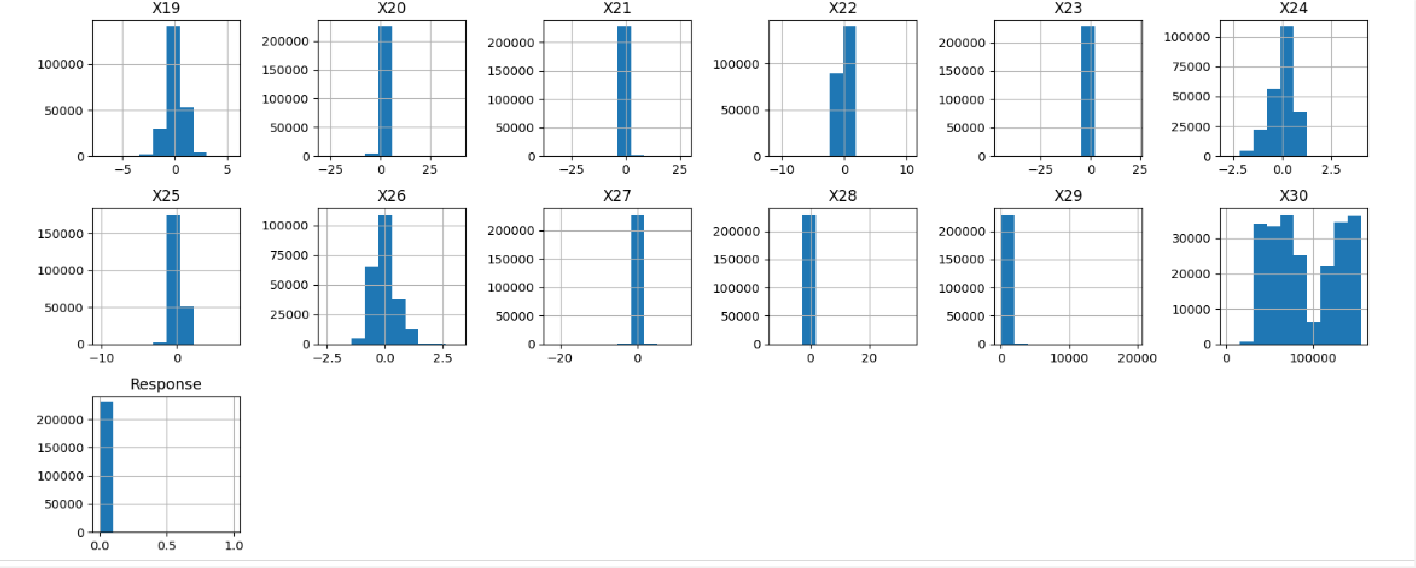
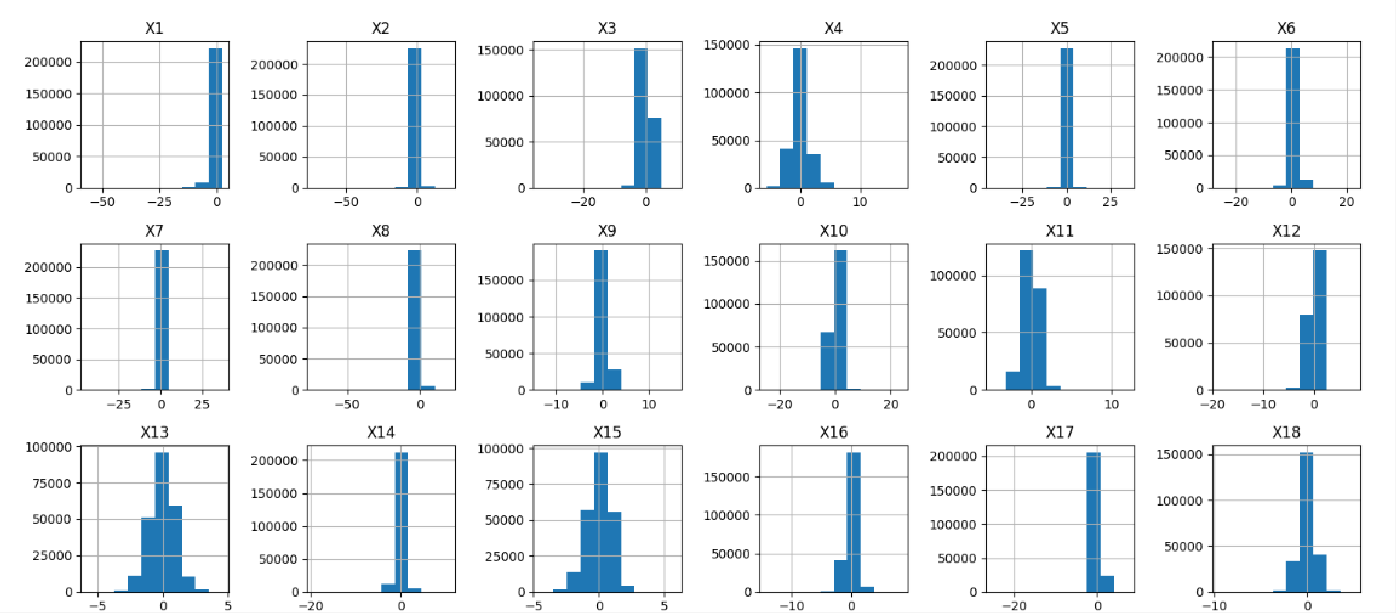
# Project Description:

Given a data of size 2,30,000 with 30 predictor variables where we have to determine whether the ball bearing is defective or non-defective. This is a classification problem where the response variable has two attributes (0,1)

0 = Non defective, 1= Defective

# Data Visualization:

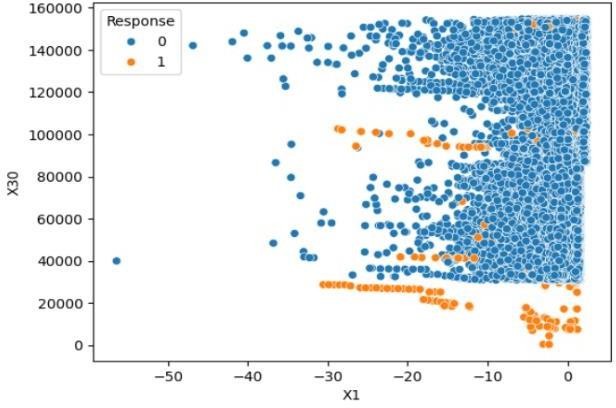
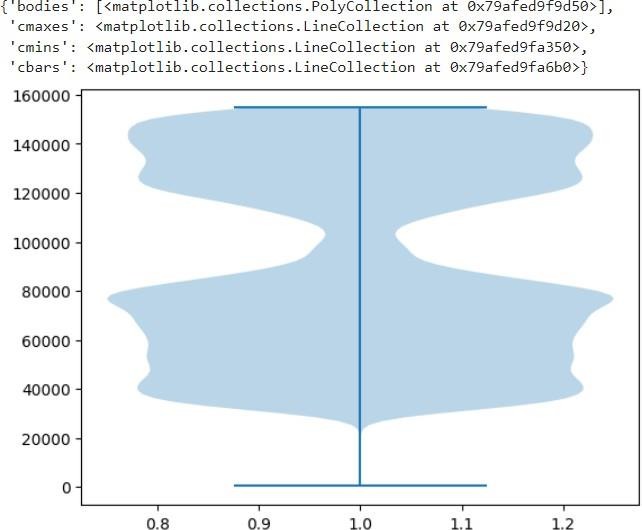
**Histogram**

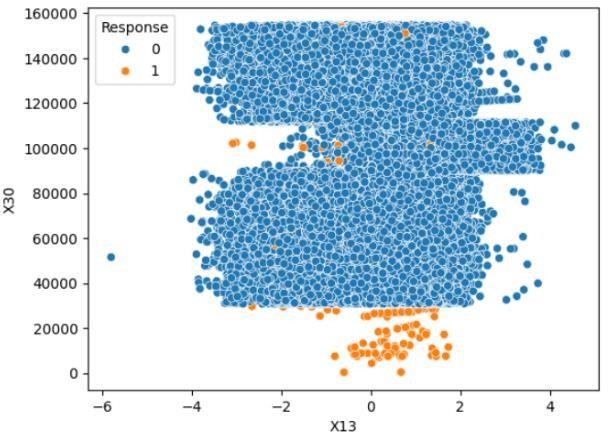
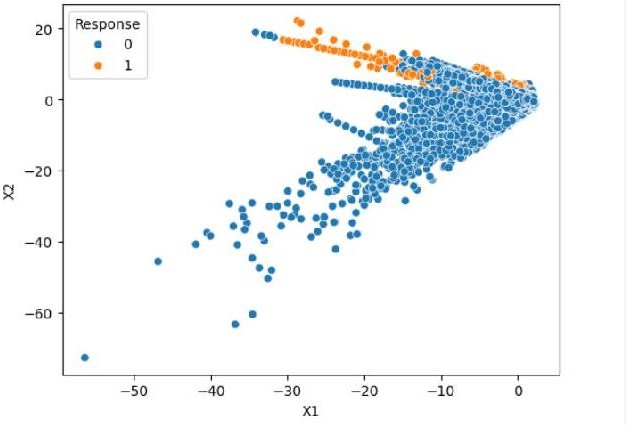


By looking at the above histogram we can infer that most of the data is distributed between the range of -1 to

1. Only predictor X30 has its data spread across the graph. This could potentially mean X30 more influential in determining the prediction outcome compared to other predictors.

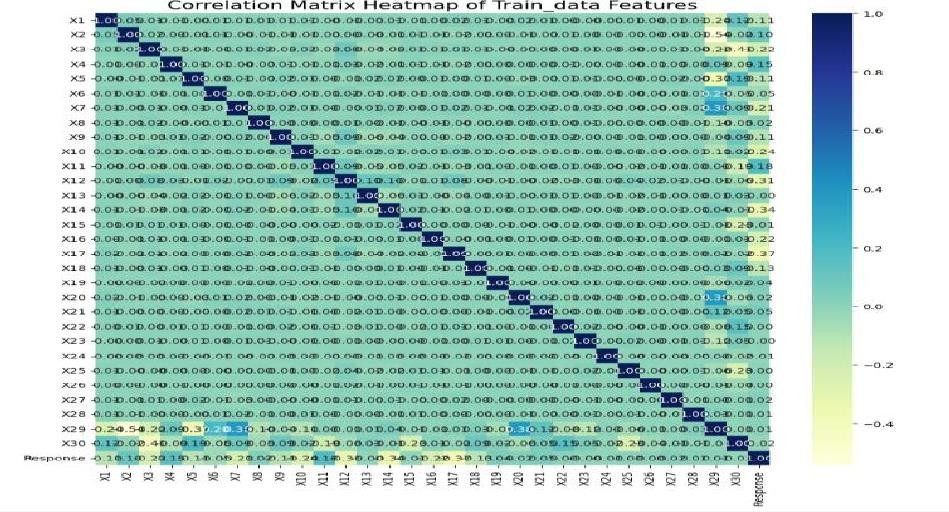
# Scatter Plot and Violin Plot:



From the above scatter plot there is a strong relationship between other predictors and X30. We can also see the X30 data points distribution from the violin plot.

# Co relation-Heat Map:

* The co-relation between all the predictors ranges from -0.1 to 0.2. Only X29 and X30 has a bit higher correlation when compared to others.
* Lower values of correlation tells us that given predictors have weak linear relationship between each other.
* We can conclude that the given predictors are Independent.

# Data Pre-Processing

Observations:

* The given data has no null values.
* The value of each predictor ranges greatly. The minimum value is -72.715728 and maximum value is 154674.000000. Machine learning models are sensitive to the scale of values used. This creates biased model and predictor with large scale will influence the outcome more.

# Standardization of Data

* By applying Standard Scalar we standardize the data. This technique reduces the scaling difference between predictors and helps reduce bias.

# Observed data problem

* In the given Training data, we have observed a problem that will cause an serious prediction error.
* Total number of Non defective ball bearings is 229541, Total number of defective ball bearings = 458.
* This is a very huge difference of data points between two classes. The model may predict the non-defective ball bearings correctly but it fails to read the patterns of Defective ball bearings

# Data re-sampling

We are using K-Fold cross validation method. It splits the data into k groups and shuffles the data randomly. This shuffling makes sure no feature is lost during data split for training.

Every observation is used while predicting.

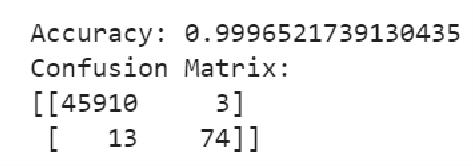
* **Data splitting:** We split the 80% of data for training and 20% for testing. It is a general procedure

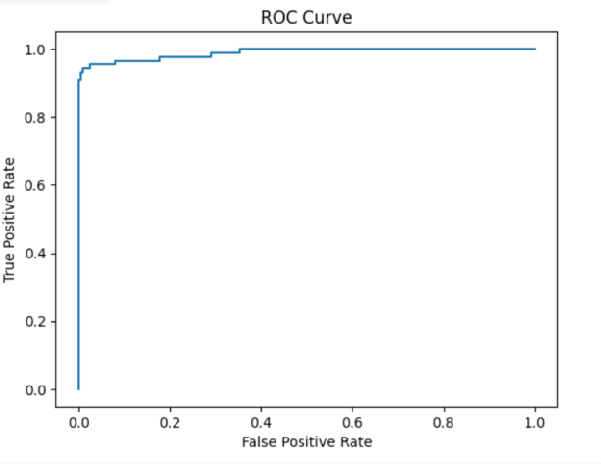
# Model Selection

* Since we have large data, significantly less attributes compared to data points and we need the classification between only two classes (i.e 0 or 1 ), we can consider Logistic Regression and Decision Tree models.
* Logistic Regression and Decision Tree models are suitable choices for our dataset due to their simplicity, interpretability, and ability to handle binary classification tasks efficiently. But each has its own advantages and disadvantages. We can experiment with both models and evaluate their performance using appropriate metrics to determine the most effective approach for our specific use case. Additionally, we may consider ensemble methods like Random Forests or Gradient Boosting as alternatives to Decision Trees for potentially improved performance.

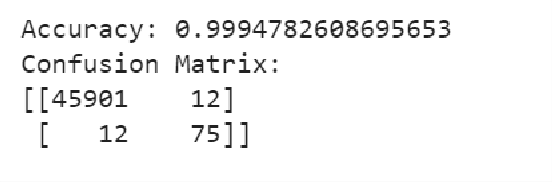
# Performance Metrics:

**Logistic Regression**



* We achieved a very high accuracy of 99%
* Specificity = 0.99993
* Simplicity = 0.85057
* Though we achieved a high accuracy if we look at confusion matrix we can clearly see that there is a significant error in predicting the Defective ball bearings.
* Hence it is not a recommendable model.

# Decision Tree



* Specificity = 0.99971
* Simplicity = 0.85057
* Similarly we achieved a very high accuracy of 99%, but looking at other performance metrics we clearly missing to capture the patterns of class 1.

# Conclusion

In conclusion, we can say that the dataset is highly imbalanced, meaning one class is much more prevalent than the other, accuracy alone may not be a reliable measure of model performance. Looking at the number of false positives and false negatives relative to true positives and true negatives. The model is making a significant number of errors, particularly false positives or false negatives, hence both models are not performing well.

# Model Selection

* Since we have large data, significantly less attributes compared to data points and we need the classification between only two classes (i.e 0 or 1 ), we can consider Logistic Regression and Decision Tree models.
* Decision Tree and logistic models are suitable choices for our dataset due to their simplicity, interpretability, and ability to handle binary classification tasks efficiently. But each has its own advantages and disadvantages. We can experiment with both models and evaluate their performance using appropriate metrics to determine the most effective approach for our specific use case. Additionally, we may consider ensemble methods like Random Forests or Gradient Boosting as alternatives to Decision Trees for potentially improved performance.

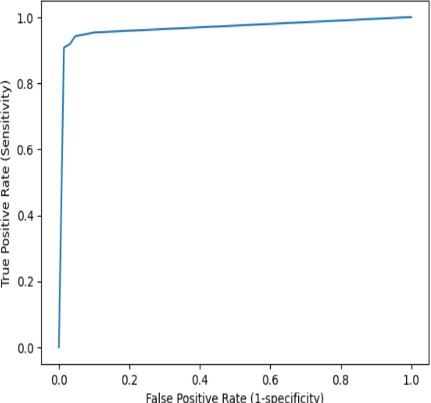
# Resampling Techniques :

In last deliverable we used cross validation resampling method. In deliverable-2 we are using *Random over and under sampling method*. In order to achieve a balanced class distribution, the Random Over Sampler approach duplicates examples from the minority class at random. The performance of classifiers is enhanced, especially for minority class prediction, by increasing the number of examples in the minority class, which helps to mitigate the imbalance issue. Another method for addressing class imbalance in machine learning datasets is random under sampling. In order to achieve balance in the dataset, Random Under Sampling lowers the number of instances in the majority class while Random Over Sampling increases the number in the minority class.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance Metric :** |  |  |  |
| **Decision Tree** |  |  |
| a) with random over sampler |  | b) with random under sampler |
| Accuracy – 96.777% |  | Accuracy – 95.520% |  |
| Specificity – 0.999 |  | Specificity – 0.999 |  |
| Sensitivity – 0.9195 |  | Sensitivity – 0.931 |  |
| Confusion matrix – [ [ 44438 | 1475 | Confusion matrix – [ [ 43712 2201 |  |
| 7 | 80]] | 6 81]] |  |

# ROC Curve

# 

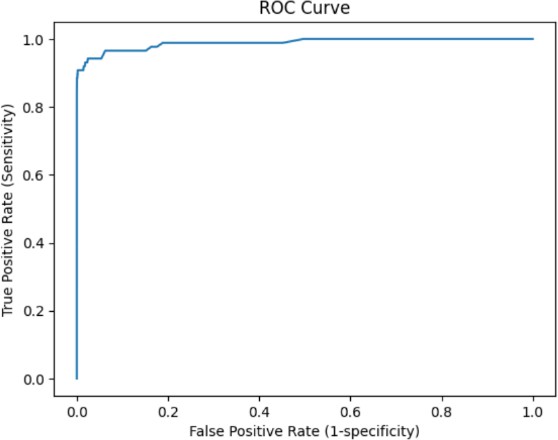
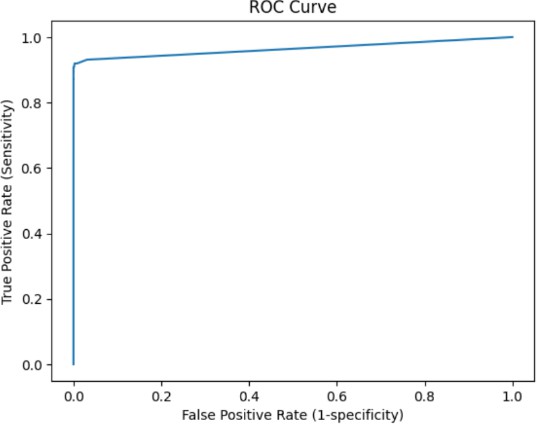


**Random Forest Methods :**

The Random Forest algorithm is an effective machine learning tree learning technique. It creates a large number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set in order to measure a random subset of characteristics in each partition. The randomization makes each tree more varied, which reduces the possibility of overfitting and improves prediction performance overall. The algorithm uses the average of each tree's output (for regression tasks) or votes (for classification tasks) to make predictions. With the help of multiple trees insights, this cooperative decision-making process produces precise and consistent results.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance Metric :** |  | | |
| **Random forest method** |
| a) with random over sampler |  |  | b) with random under sampler |
| Accuracy – 99.999% |  |  | Accuracy – 95.520% |
| Specificity – 0.999 |  |  | Specificity – 0.999 |
| Sensitivity – 0.8735 |  |  | Sensitivity – 0.931 |
| Confusion matrix – [ [ 45911 | 2 |  | Confusion matrix – [ [ 45000 913 |
| 11 |  | 76]] | 6 81]] |

# ROC Curve :



After carefully observing the accuracy, confusion matrix we can state that random forest over sample method is good efficient compare to the decision tree. As we know that there is imbalance in the dataset, we just resolved that by using the random resampling techniques. However it has its own disadvantages, These resampling techniques involves randomly duplicating or randomly removing data points. However, it may lead to overfitting or loss of important patterns. This sampling method introduces bias and hence it decreases the model performance.

In the next deliverable we will resolve this by using hybrid resampling method.

# Resampling Technique:

The resampling technique that combines slight oversampling of the minority class and slight undersampling of the majority class is known as "**hybrid resampling**" or "balanced resampling". It aims to achieve a balance between the two classes by reducing the size of the majority class and increasing the size of the minority class, thereby reducing the risk of overfitting and addressing class imbalance.

Different methods can be used to implement this hybrid approach, including:

* **Synthetic Minority Over-sampling Technique (SMOTE)**, which generates synthetic samples for the minority class.
* **Random undersampling**, which randomly removes samples from the majority class.
* **NearMiss**, are the other advanced techniques for balancing data while maintaining important features and relationships within the dataset.

Combining these techniques can be effective in creating a more balanced dataset for training machine learning models, which in turn can improve classification accuracy and reduce bias.

After applying Hybrid resampling technique the results are:

# Decision Tree:

Accuracy: 0.9936086956521739

Sensitivity: 0.8695652173913043

Specificity: 0.9997370612209124

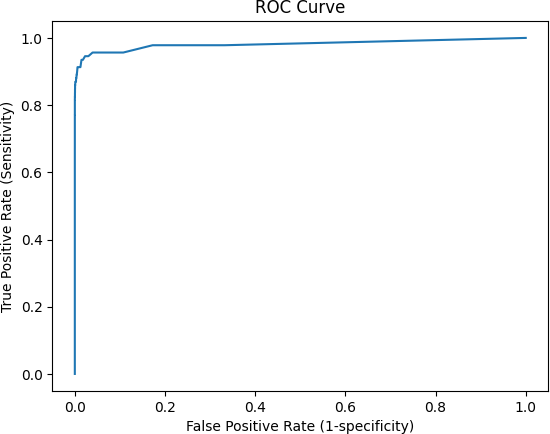
Confusion Matrix:

[[45626 282]

[ 12 80]]

# Random Forest:

|  |  |  |
| --- | --- | --- |
| Accuracy: | 0.9992826086956522 | Sensitivity: 0.8369565217391305 |
|  |  | Specificity: 0.9996732382093454 |
|  |  |  |
| Confusion | Matrix: |  |
| [[45890 | 18] |  |
| [ 15 | 77]] |  |



As we can observe the results have improved significantly. The False Positive and False Negitive of the matrix has decreased a lot. But it looks like it over fits the data, when we apply on unseen data a lot of variance can be found.

**Model Selection SVM**

Support Vector Machines (SVM) are a class of supervised learning algorithms used for classification, regression, and outlier detection. They are particularly known for their robustness in handling high-dimensional data and achieving high accuracy, especially in binary classification tasks.

Accuracy: 0.9855434782608695 Sensitivity: 0.9130434782608695

Specificity: 0.9998232395766588

Confusion Matrix: [[45251 657]

[ 8 84]]

**Section 4: Discussion & Conclusions**

**Decisions Made**

* **Resampling Techniques**: Used hybrid resampling (combination of oversampling and undersampling) to address the severe class imbalance in the dataset.
* **Model Selection**: After evaluating Logistic Regression, Decision Trees, Random Forests, and SVM, we finalized SVM due to its superior sensitivity and specificity. While its accuracy was slightly lower, it performed well in detecting defective ball bearings, as indicated by its confusion matrix.
* **Performance Metrics**: Sensitivity and specificity were prioritized over overall accuracy to ensure the model effectively handled the minority class.

**Difficulties Faced**

* **Class Imbalance**: The disproportionate representation of non-defective samples (majority class) made it challenging for models to correctly predict defective ball bearings (minority class).
* **Overfitting in Resampling**: Techniques like random oversampling occasionally introduced repetitive data, reducing generalization on unseen datasets.
* **Complexity of Decision Trees and Random Forests**: While these models showed high accuracy, they failed to consistently perform well in terms of sensitivity and specificity for the minority class.

**Things That Worked**

* **Hybrid Resampling**: This technique significantly improved minority class detection by creating a more balanced training dataset.
* **SVM Model**: Delivered strong sensitivity and specificity, offering a more balanced and reliable performance despite slightly lower accuracy.
* **Data Standardization**: Scaling the predictors minimized bias due to varying magnitudes, ensuring fair treatment of all variables.

**Things That Didn’t Work Well**

* **Logistic Regression and Decision Trees**: These models lacked the complexity to handle the nuances of the dataset, resulting in poor performance for the minority class.
* **Random Oversampling**: Introduced repetitive data points, which occasionally led to overfitting, reducing test performance.

**Conclusion**

* SVM was selected as the final model due to its strong sensitivity and specificity, making it highly reliable for predicting defective ball bearings. While its accuracy was slightly lower, it achieved the best results for the minority class, as shown by its confusion matrix.
* Future work could include fine-tuning the SVM model further or experimenting with advanced techniques like kernel optimization to enhance performance.

**Section 5: Project Plan / Task Distribution**

**Task Distribution**

1. **Data Preprocessing**: Managed by **Dedeepya Vesangi**
   * Tasks:
     + Standardized predictors using Standard Scaler to handle scale variations.
     + Addressed class imbalance using resampling techniques like oversampling, undersampling, and hybrid resampling.
2. **Model Implementation and Tuning**: Led by **Nithin Aleti**
   * Tasks:
     + Implemented machine learning models, including Logistic Regression, Decision Trees, Random Forest, and SVM.
     + Experimented with hyperparameters for SVM to improve sensitivity and specificity.
     + Evaluated models using confusion matrices, sensitivity, and specificity.
3. **Performance Analysis and Reporting**: Conducted by **Harshitha Manaswini Vadavalli**
   * Tasks:
     + Analyzed results using confusion matrices, sensitivity, specificity, and accuracy.
     + Created visualizations like histograms, scatter plots, and violin plots to support findings.
     + Documented the results and compiled the final report.

**Justification**

Each team member contributed to distinct components of the project based on their skills and expertise. This division of responsibilities ensured a thorough exploration of models and techniques, leading to a well-rounded and effective solution.