# **Defects Repair Cost Prediction – Project Report**

## **1. Project Introduction**

### **1.1 Overview**

This project aims to analyze manufacturing defect data to understand patterns and predict the **repair cost** associated with different types of defects.  
 The objective is to help manufacturers identify **high-cost defect categories** and improve **production quality** through data-driven insights.

### **1.2 Problem Statement**

Defect repairs can lead to significant cost implications for a company. By analyzing historical data and predicting repair costs, we can enable better budgeting and process optimization.

### **1.3 Objectives**

* Perform Exploratory Data Analysis (EDA) to find key trends.
* Identify factors affecting repair costs.
* Build machine learning models to predict repair costs accurately.
* Provide actionable insights and recommendations.

### **1.4 Dataset Description**

* **Dataset Name:** Defects.csv
* **Total Records:** 1000
* **Total Features:** 8
* **Target Variable:** repair\_cost
* **Data Source:** Provided on Kaggle

## 

## **2. Data Understanding & Preprocessing**

### **2.1 Data Overview**

The dataset contains columns such as:

* defect\_id
* product\_id
* defect\_date
* defect\_type
* defect\_location
* severity
* inspection\_method
* repair\_cost

### **2.2 Data Cleaning**

* Removed unnecessary column: Unnamed: 0
* Checked for null values and duplicates
* Converted defect\_date to **datetime format**

### **2.3 Feature Encoding**

### Categorical features were encoded using **Label Encoding** for model compatibility. Example mappings:

defect\_type: {'Cosmetic': 0, 'Functional': 1, 'Structural': 2}

defect\_location: {'Component': 0, 'Internal': 1, 'Surface': 2}

severity: {'Critical': 0, 'Minor': 1, 'Moderate': 2}

inspection\_method: {'Automated Testing': 0, 'Manual Testing': 1, 'Visual Inspection': 2}

## **3. Exploratory Data Analysis (EDA)**

### **3.1 Univariate Analysis**

* Visualized the frequency of defect types and severity levels using bar charts.
* Analyzed the distribution of repair costs to detect skewness.

### **3.2 Bivariate Analysis**

* Compared average repair cost by defect type, location, and severity.
* Observed that *Structural defects* and *Critical severity* had the highest repair costs.

### **3.3 Key Insights**

* Most defects were **Functional** and **Moderate** in severity.
* Repair costs were higher for **Internal defects** compared to Surface-level ones.
* **Manual inspection** was linked to slightly higher costs compared to automated testing.

## **4. Model Building & Evaluation**

### **4.1 Data Splitting**

Data was split into training and testing sets:

* Train: 80%
* Test: 20%

### **4.2 Models Used**

| Model | MAE | MSE | RMSE | R² Score |
| --- | --- | --- | --- | --- |
| Linear Regression | 258.89 | 88639.95 | 297.72 | -0.0044 |
| Random Forest Regressor | 260.77 | 91584.58 | 302.62 | -0.0378 |
| Gradient Boosting Regressor | 259.74 | 91390.38 | 297.72 | -0.0356 |

### **4.3 Interpretation**

* The Linear Regression model performed slightly better than Random Forest and Gradient Boosting, likely due to the small dataset size (1000 rows).
* Negative R² values indicate limited predictive power — additional data or engineered features could improve performance.

## **5. Insights & Recommendations**

### **5.1 Key Insights**

* The cost variation depends largely on **defect severity** and **defect type**.
* Automated inspection methods are more consistent and cost-effective.
* Some defect categories may need preventive maintenance to avoid costly repairs.

### **5.2 Recommendations**

* **Automate defect inspection** wherever possible to reduce manual errors.
* **Focus on high-cost categories** (Structural + Critical defects).
* **Expand dataset** with production line parameters for deeper learning models.
* Implement **cost monitoring dashboards** to track performance over time.

## **6. Future Work**

* Experiment with advanced models like **XGBoost**, **LightGBM**, or **Neural Networks**.
* Apply **feature engineering** such as:  
  + Interaction features between defect type and severity
  + Cyclical encoding of month from defect\_date
* Perform **cross-validation** with higher folds for more reliable evaluation.
* Integrate **Power BI or Streamlit dashboards** for interactive visual reporting.

## **7. Conclusion**

This project successfully demonstrates a complete data science pipeline — from data understanding and cleaning to modeling and insight generation.  
 Although model performance was limited due to data size, the process offers a scalable foundation for predictive maintenance and cost optimization.

## **8. Contact**

**Author:** Bhupendra Shivhare ( Aspiring Data Scientist )  
 **Email:** *[shivharebhupendra0@gmail.com]* **LinkedIn:** *[https://www.linkedin.com/in/bhupendra-shivhare-a8a02a25b/]* **GitHub Repository:** *[https://github.com/shivharebhupendra/Defect-Prediction-Root-Cause-Study/tree/main]*