# Machine Failure Prediction Using Sensor Data

## 🎯 Objective

The goal of this project is to develop a machine learning model that predicts potential machine failures using real-time sensor data, helping enable predictive maintenance and reduce unexpected downtime.

## 📂 Dataset Overview

• Source: Provided for capstone project  
• Total records: 944 rows  
• Total features: 9 input features + 1 target (fail)  
• Target Variable: fail (1 = Failure occurred, 0 = Normal operation)

## 🔍 Features Used

|  |  |
| --- | --- |
| Feature | Description |
| footfall | Number of people/objects passing the machine |
| tempMode | Machine’s temperature setting |
| AQ | Air Quality Index near the machine |
| USS | Ultrasonic sensor data (distance/proximity) |
| CS | Electrical current usage (Current Sensor) |
| VOC | Volatile organic compounds detected |
| RP | Rotational position (or RPM) |
| IP | Input pressure to the machine |
| Temperature | Operating temperature |

## 🧠 Tools & Technologies

• Python  
• Pandas  
• Matplotlib & Seaborn  
• Scikit-learn  
• RandomForestClassifier  
• Joblib

## 🔄 Project Workflow

1. Data Loading – Sensor data is read from 'data (1).csv'  
2. Exploratory Analysis – Check for nulls, print sample, view heatmap  
3. Data Preparation – Features/target split, train-test split  
4. Model Training – Random Forest trained on 80% of data  
5. Evaluation – Confusion matrix, classification report printed  
6. Outputs Saved – predictions.csv, machine\_failure\_model.pkl  
7. Feature Importance – Visualized with bar chart

## 📈 Model Performance (Example)

Accuracy: 93%  
Precision: 88%  
Recall: 75%  
F1-Score: 81%

## ✅ Deliverables

• main.py – Python script  
• data (1).csv – Dataset  
• predictions.csv – Model predictions  
• machine\_failure\_model.pkl – Trained model  
• Project Overview document

## 📌 Future Improvements

• Try advanced models (XGBoost, LightGBM)  
• Hyperparameter tuning  
• Deploy as a web app (Streamlit or Flask)  
• Enable auto-retraining and dashboard