**Title Page**

* **Title**: Anoma Data
* **Subtitle**: Detailed Project Documentation
* **Author**: [Ajay Parmar]

**Table of Contents**

* Description
* Problem Statement
* Data Sources
* Code and Models
  + Code Overview
  + Models Used
* Instructions
  + Installation
  + Setup
  + Running the Code
  + Testing
* Examples
* Acknowledgements

**Description** This project focuses on detecting anomalies in a dataset using machine learning techniques. The goal is to build an effective anomaly detection model that can identify unusual patterns in the data, which could be crucial for various applications such as fraud detection, network security, and fault detection in systems.

**Problem Statement** Many different industries need predictive maintenance solutions to reduce risks and gain actionable insights through processing data from their equipment.

Although system failure is a very general issue that can occur in any machine, predicting the failure and taking steps to prevent such failure is most important for any machine or software application.

Predictive maintenance evaluates the condition of equipment by performing online monitoring. The goal is to perform maintenance before the equipment degrades or breaks down.

This Capstone project is aimed at predicting the machine breakdown by identifying the anomalies in the data.

The data we have contains about 18000+ rows collected over few days. The column ‘y’ contains the binary labels, with 1 denoting there is an anomaly. The rest of the columns are predictors.

**Data Sources**

* **File Path**: C:\Users\parma\Downloads\AnomaData (5).xlsx
* **Data Columns**: Includes various feature columns and a 'time' column.
* **Target Variable**: 'y', which indicates the presence (1) or absence (0) of anomalies.

**Code and Models**

**Code Overview**

1. **Data Loading and Exploration**:
   * Load the dataset using pandas.
   * Explore dataset with basic statistics and visualizations:
     + Histograms to show feature distributions.
     + Box plots to identify outliers.
     + Correlation matrix to visualize relationships between features.
2. **Data Preprocessing**:
   * **Missing Values**: Imputed with median values.
   * **Outlier Detection**:
     + Identified using Z-scores.
     + Outliers filtered using the IQR method.
3. **Feature Engineering**:
   * Standardize features using StandardScaler.
   * Create interaction terms and polynomial features.
   * Apply PCA for dimensionality reduction.
4. **Model Selection and Training**:
   * **Model**: OneClassSVM for anomaly detection.
   * Train-Test Split: Split data into training and testing sets.
5. **Evaluation**:
   * Classification reports for model performance.
   * ROC-AUC score to measure the model’s ability to distinguish between classes.
6. **Model Saving**:
   * Save the trained model using joblib.

**Models Used**

* **OneClassSVM**: A model specifically designed for anomaly detection. It works by finding a decision boundary that separates the majority of the data from the anomalies.

**Instructions**

**Installation** Install the required libraries:

bash

Copy code

pip install pandas numpy seaborn matplotlib scikit-learn scipy joblib

**Setup**

1. Ensure the dataset AnomaData (5).xlsx is located at the specified path.
2. Make sure all dependencies are installed.

**Running the Code** Execute the Python script:

bash

Copy code

python your\_script\_name.py

Replace your\_script\_name.py with the name of your Python file.

**Testing** Review outputs:

* Visualizations of feature distributions, outliers, and correlations.
* Classification reports and ROC-AUC scores.

**Examples** The script generates:

* Histograms of feature distributions.
* Box plots for outlier detection.
* Correlation heatmap.
* Time series plot for anomalies.
* Histograms comparing feature distributions for anomalies vs. non-anomalies.

**Acknowledgements**

* Libraries and tools used: pandas, numpy, seaborn, matplotlib, scikit-learn, scipy, joblib.
* Reference any specific tutorials or resources used.