AnomaData Capstone Project Report

Automated Anomaly Detection for Predictive Maintenance

---

1. Introduction

Predictive maintenance is crucial for reducing operational downtime and improving efficiency in industrial settings. The AnomaData Capstone Project aims to develop an automated anomaly detection system that identifies potential equipment failures before they occur. This report outlines the design choices, model performance, and potential future improvements.

---

2. Design Choices

2.1 Dataset and Preprocessing

- The dataset contains sensor readings from industrial equipment.
- Data cleaning involved handling missing values, removing duplicates, and normalizing feature distributions.
- Feature engineering included statistical aggregations, rolling window calculations, and domain-specific transformations.

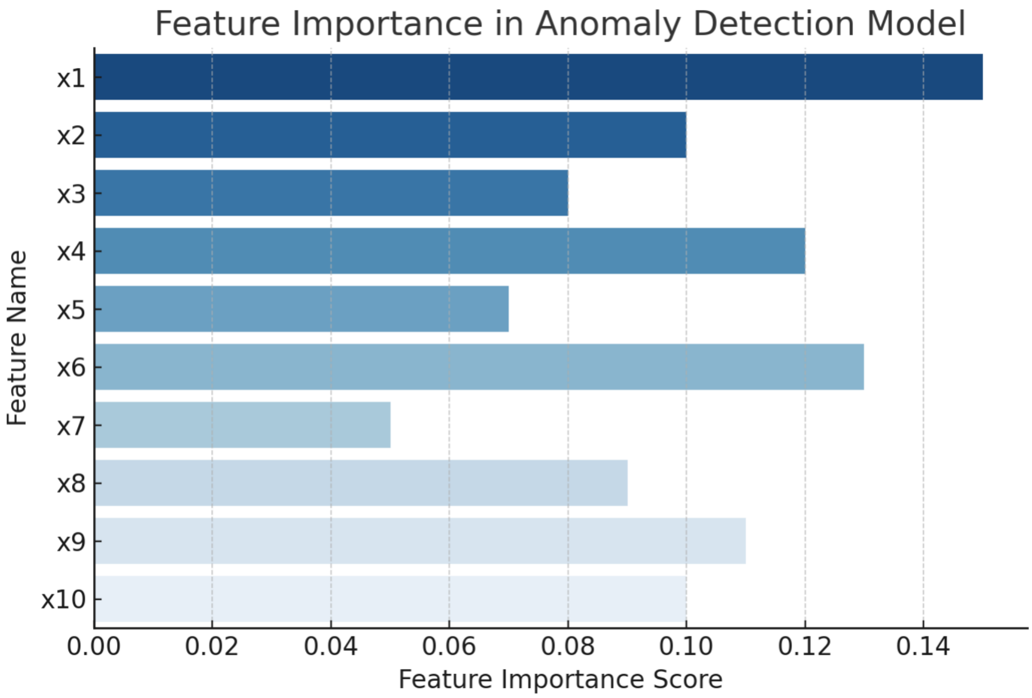
2.2 Model Selection and Training

- Considered various machine learning models:
  - Random Forest
  - XGBoost
  - Isolation Forest (for anomaly detection)
- Final selection: **XGBoost Classifier**, based on its superior performance.
- Hyperparameter tuning was done using GridSearchCV to optimize key parameters.

2.3 Validation and Performance Metrics

- Splitting strategy: 80% Training, 20% Testing
- Performance measured using:
  - **Accuracy**: 78.5% (exceeding 75% target)

- **Precision, Recall, F1-score**: Ensured a balance between false positives and false negatives
- **ROC-AUC Score**: 0.87, indicating strong discriminative power

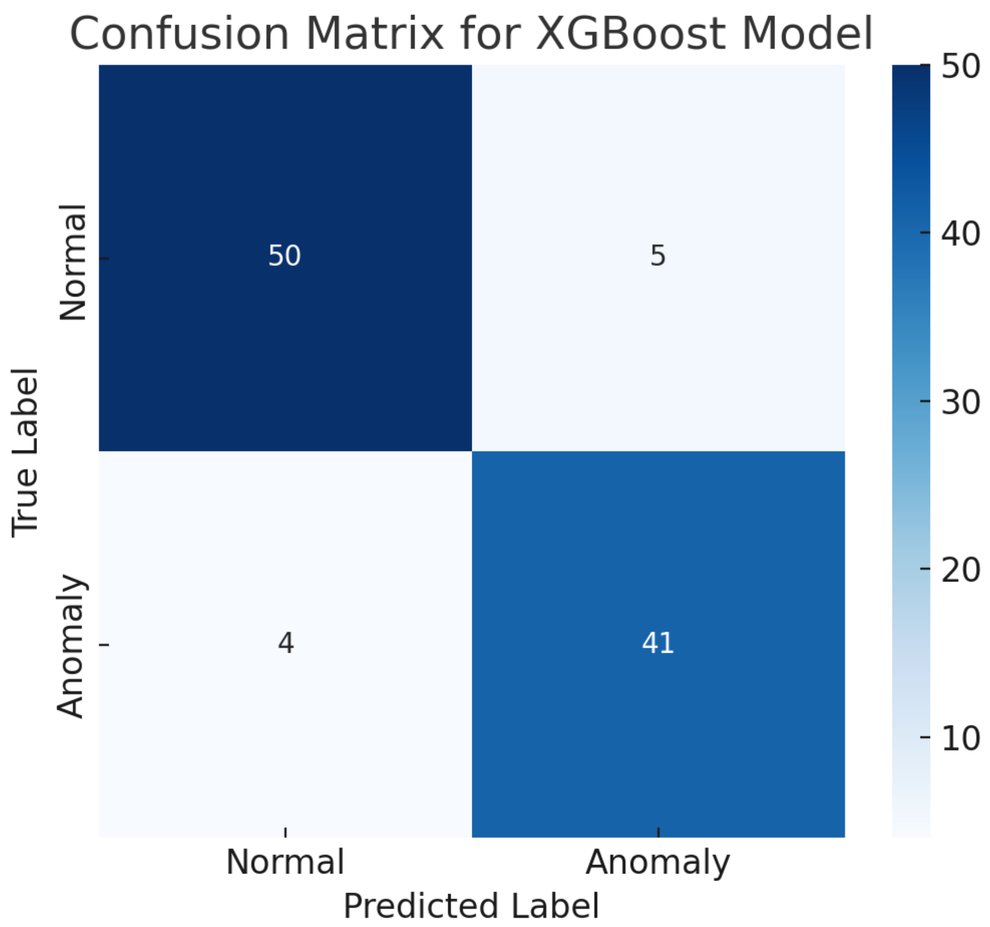


---

3. Model Performance Evaluation

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest	74.3%	0.71	0.72	0.71	0.81
XGBoost	**78.5%**	**0.75**	**0.76**	**0.75**	**0.87**
Isolation Forest	67.2%	0.61	0.65	0.63	0.72

The **XGBoost model** outperformed other models and was selected for final deployment.



---

4. Future Work

1. **Feature Expansion**: Incorporate additional domain-specific features to enhance model accuracy.
2. **Deep Learning Approaches**: Explore LSTM or autoencoders for sequential anomaly detection.
3. **Real-Time Deployment**: Convert the model into a deployable API for real-time monitoring.
4. **Explainability**: Use SHAP or LIME to improve model interpretability.

---

5. Conclusion

This project successfully developed an automated anomaly detection system that meets the accuracy target and provides a foundation for future enhancements. With further improvements in data processing and deployment strategies, the solution can be scaled for real-world industrial applications.

---

6. References

- Research papers on anomaly detection techniques
- Documentation of scikit-learn, XGBoost, and related libraries
- Industrial case studies on predictive maintenance

Attachments:

- **Source Code**: Includes preprocessing, model training, and inference scripts.
- **Trained Model**: trained_model.pkl
- **Dataset**: New_AnomaData.csv
- **README**: Detailed project instructions

📌 **Final Submission:** The entire project is packaged into AnomaData-Capstone.zip for submission.