# PREVENTIVE MAINTANENCE

## NASA WATER PUMP

APRIL 2024


# Objective

Showcase how such models can accurately predict failures, leading to proactive maintenance interventions aimed at reducing downtime and optimizing maintenance schedules.

# Business Understanding

## Goal

DEVELOP AND IMPLEMENT MODELS TO ENABLE ACCURATE FAILURE PREDICTIONS

## Goal

- REDUCED BREAKDOWNS
- EXTEND EQUIPMENT LIFE
- OPTIMIZE MAINTENANCE SCHEDULE
- ENHANCE OPERATONS EFFICIENCY

# Data Set

- Data from Kaggle
- Analyzed sensor data from NASA water pumps
- Binary Classification
- Recall measurement

# Data Insight

## ATTRIBUTES

- Total Entries: 220.320
- Total Columns: 55
- Unnamed: 0Column:ID/Index
- Sensor 15 column: Removing

## MACHINE STATUS

- Normal: 205,836
- Recovering: 14,477
- Broken: 7

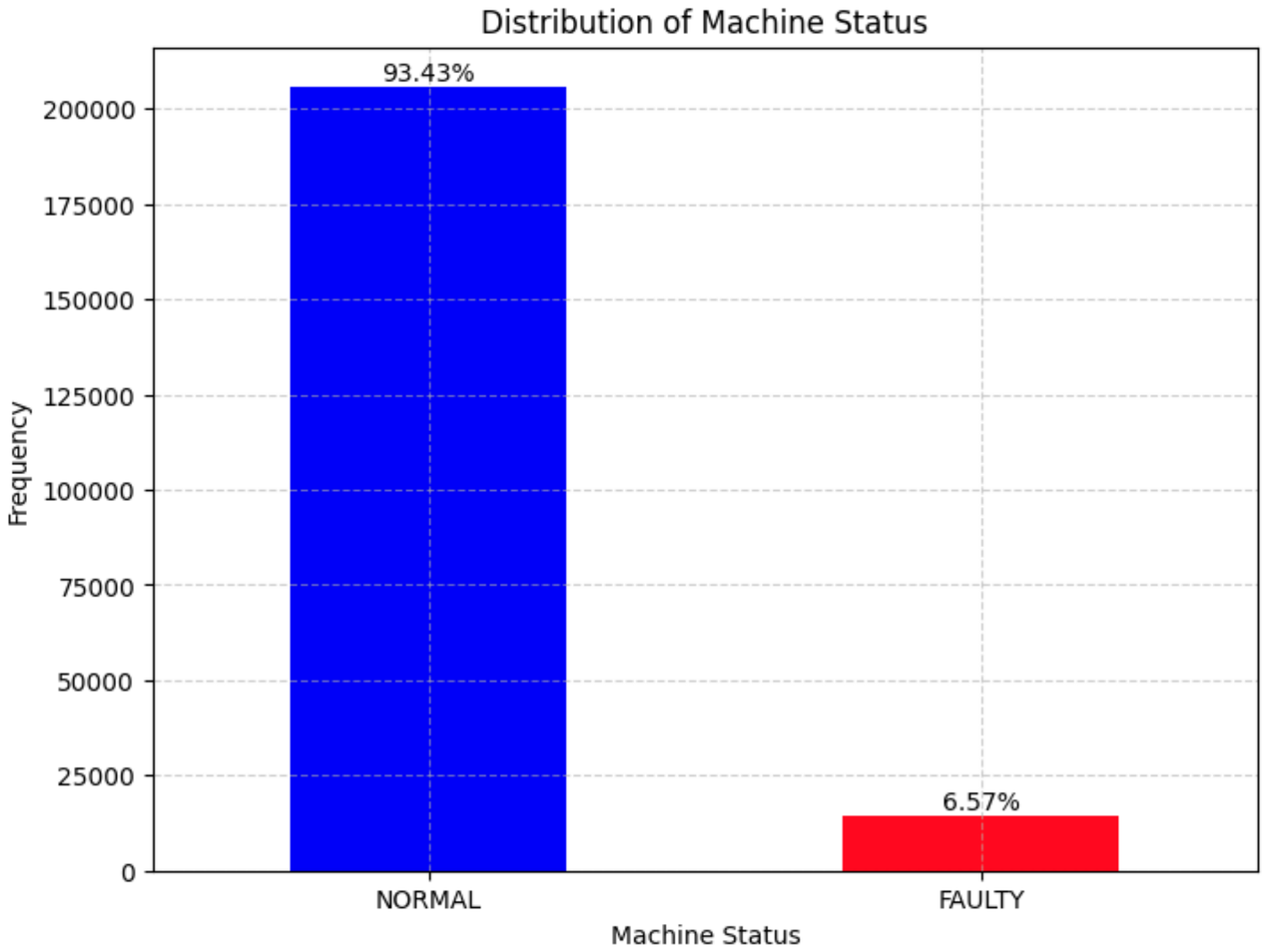
## MISSING VALUES

- 77,017 and 220,320 missing
- Checked for duplicates
- Used median or mode to fill in N/A.

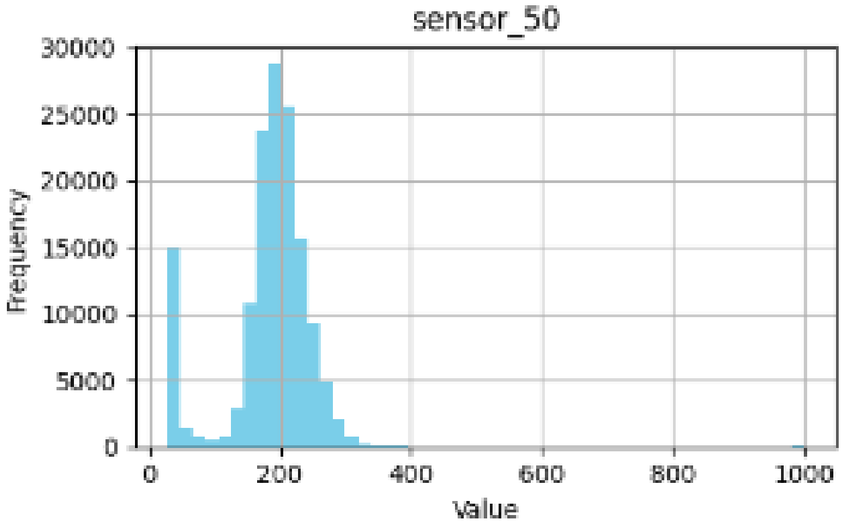
# Data Prep

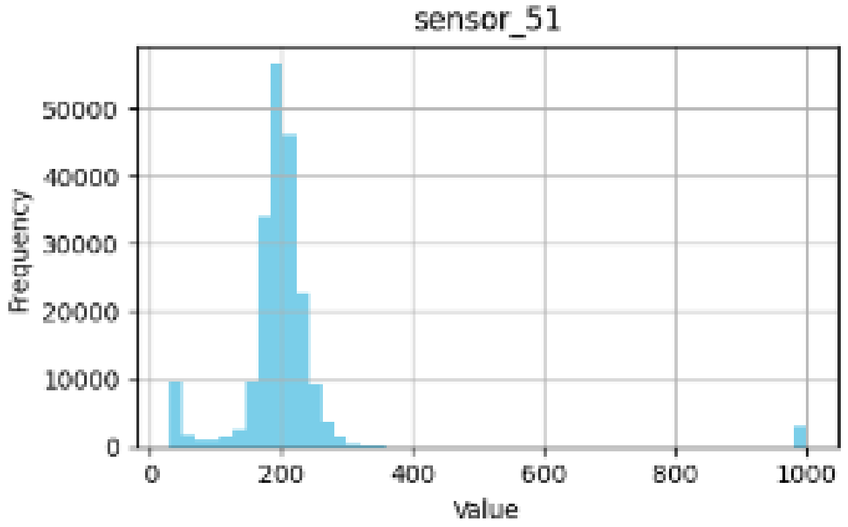
- Cleaned and preprocessed data for quality and consistency
- Merged similar machine statuses to simplify the target variable
- Visualized data distribution and outliers
- Dimension reduction(PCA)

# Data

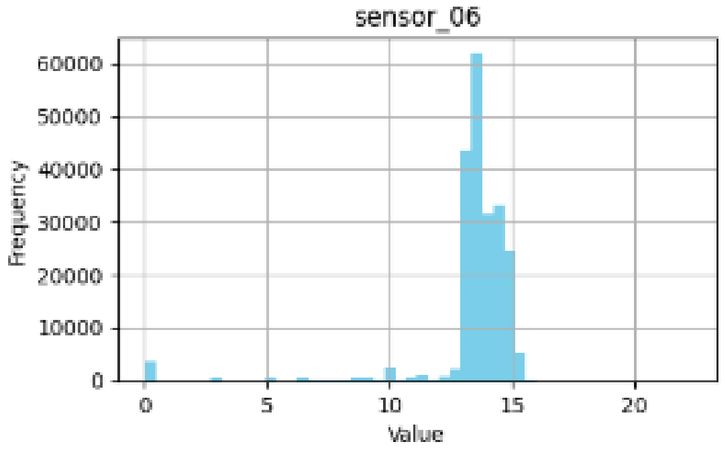

Distribution of Machine Status
93.43%
6.57%
200000
175000
150000
125000
100000
75000
50000
25000
0
Frequency
NORMAL
FAULTY
Machine Status

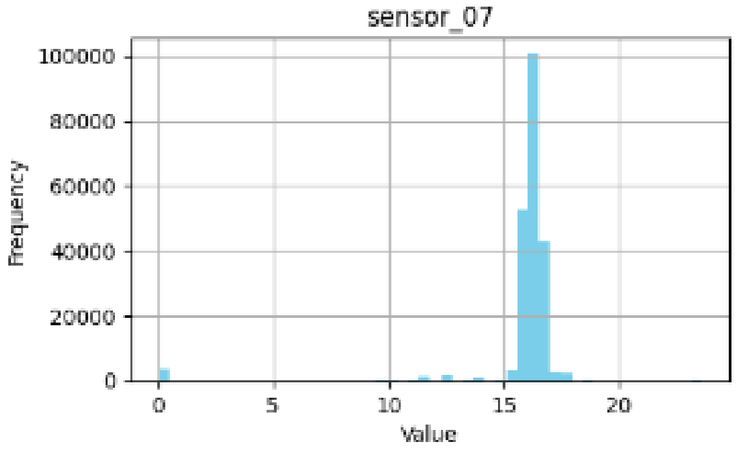
# Data

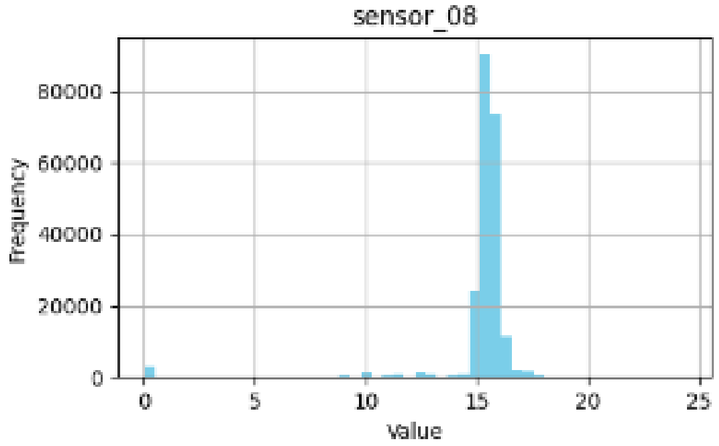

sensor_50
Frequency
Value
0
5000
10000
15000
20000
25000
30000
0
200
400
600
800
1000

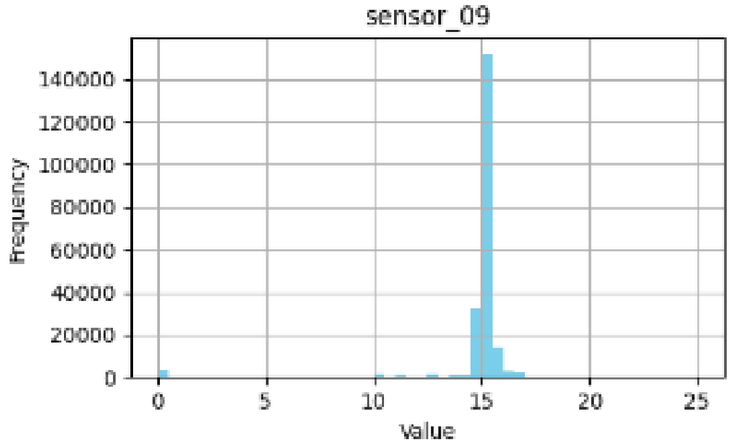

sensor_51
Frequency
Value
0
10000
20000
30000
40000
50000
0
200
400
600
800
1000

# Data


sensor_06
Frequency
Value
0
10000
20000
30000
40000
50000
60000
0
5
10
15
20


sensor_07
Frequency
Value
0
20000
40000
60000
80000
100000
0
5
10
15
20


sensor_08
Frequency
Value
0
20000
40000
60000
80000
0
5
10
15
20
25


sensor_09
Frequency
Value
0
20000
40000
60000
80000
100000
120000
140000
0
5
10
15
20
25

# Model Process


01

## Regression Models

- Logistic and Random Forest

02

## Ensemble Models

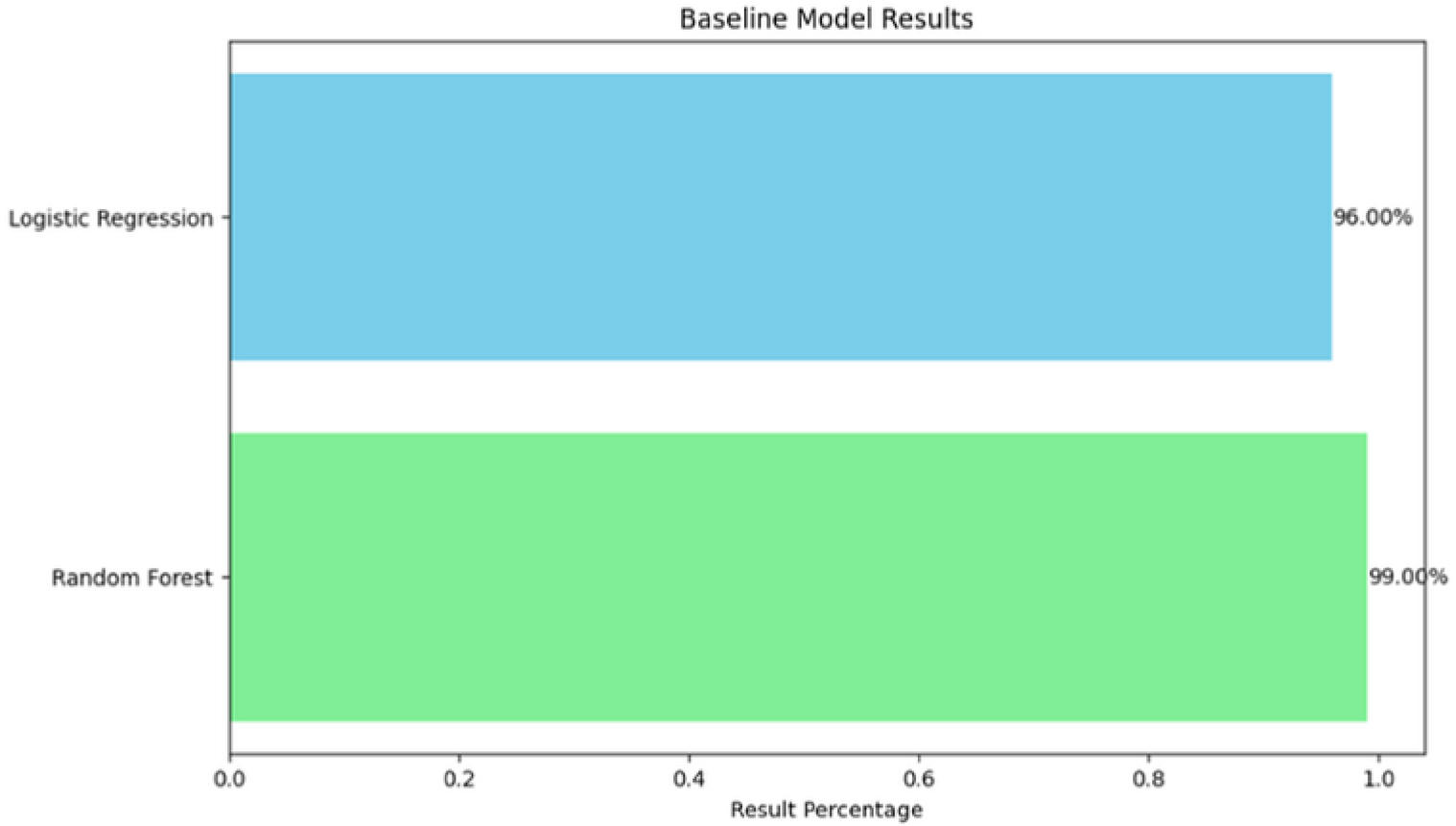
- Bagging, AdaBoost, XGBoost, Stacking, and Voting

03

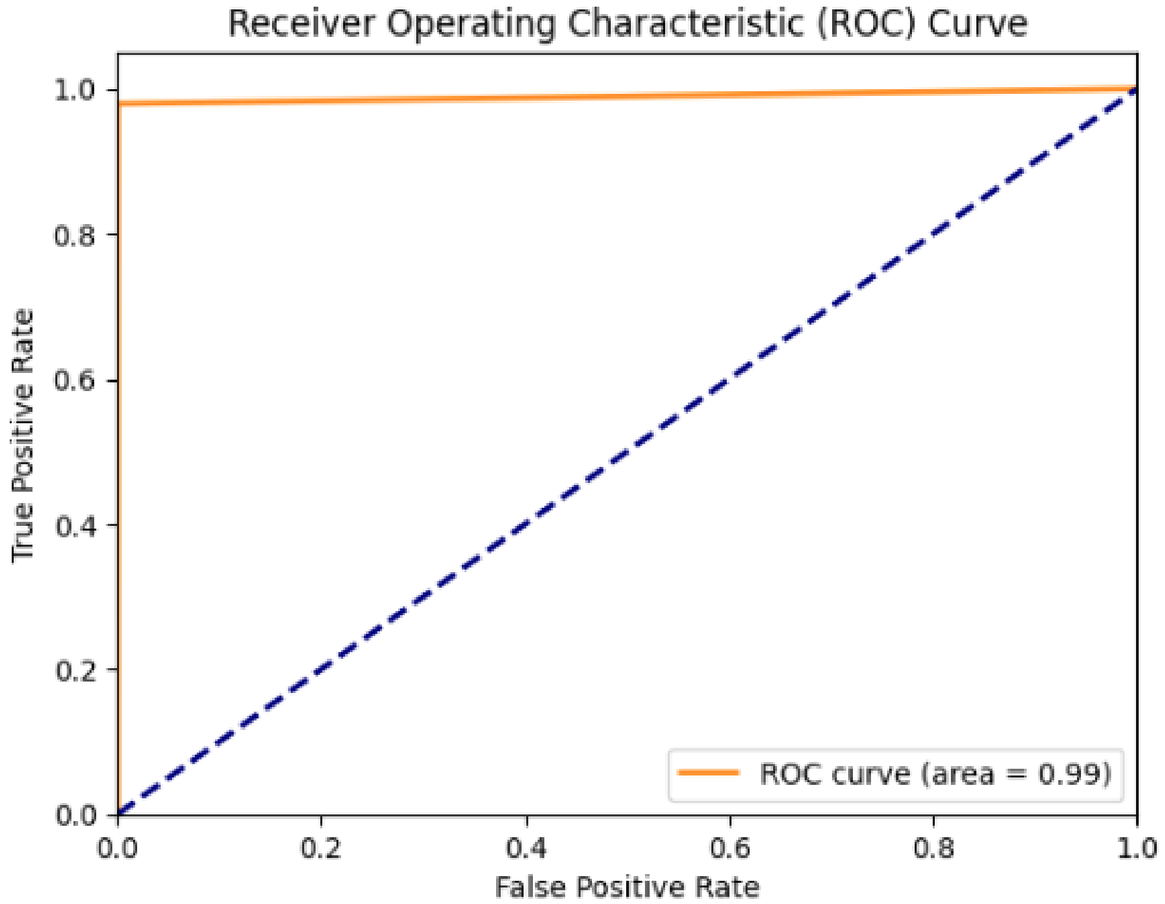
## Tuning

- Gridsearch validation
- Hyperparameter tuning

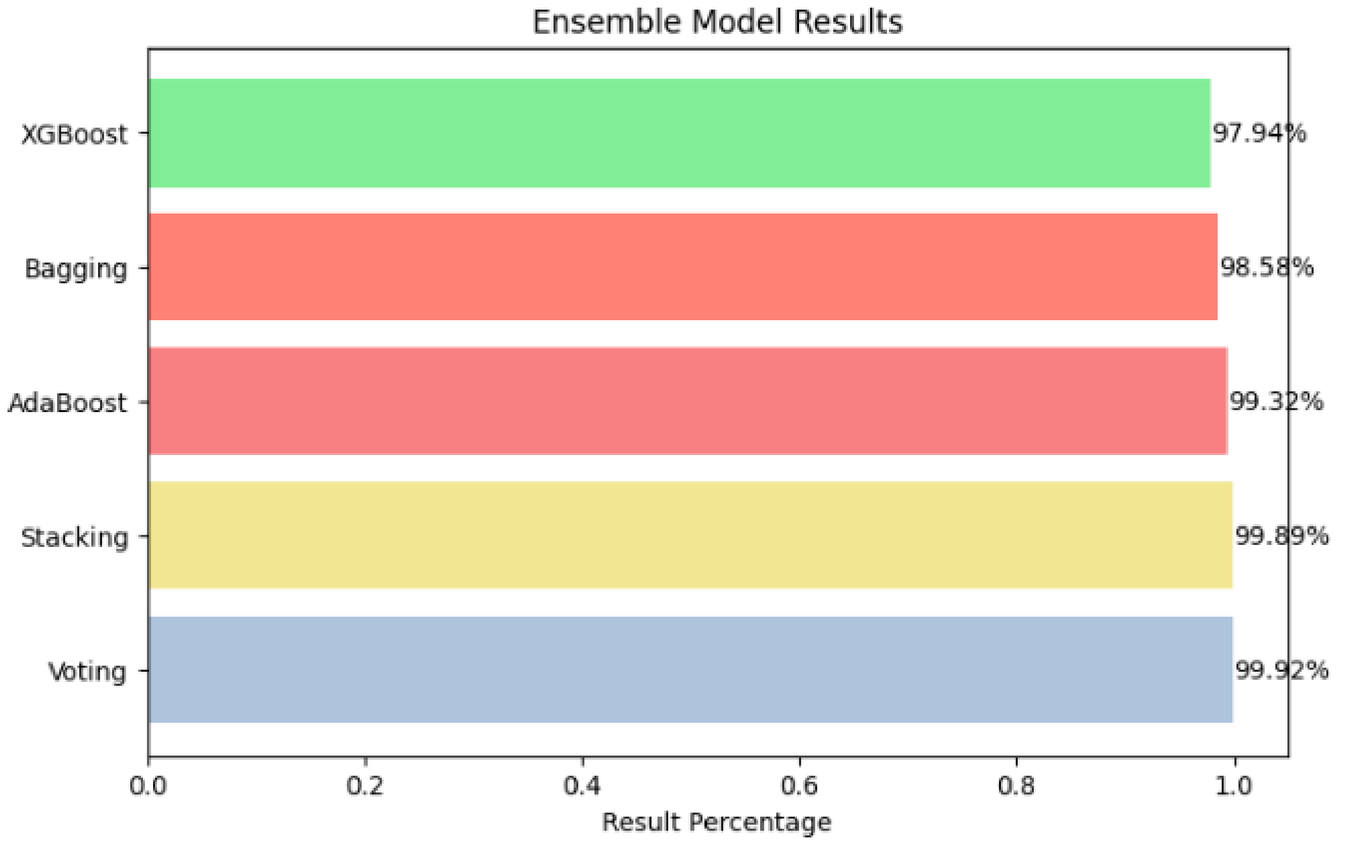
# Data


Baseline Model Results
Logistic Regression
Random Forest
96.00%
99.00%
0.0
0.2
0.4
0.6
0.8
1.0
Result Percentage

# Data


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.99)

# Data


Ensemble Model Results
XGBoost
Bagging
AdaBoost
Stacking
Voting
97.94%
98.58%
99.32%
99.89%
99.92%
0.0
0.2
0.4
0.6
0.8
1.0
Result Percentage

# Results

- Achieved high consistency and accuracy across multiple models
- Near-perfect ROC AUC scores indicating excellent model performance
-

# Next Steps

1

## Refine models

- Continuous monitoring of key sensors for real-time predictive maintenance

2

## Further experiment

- Time Series
- Feature engineering

3

## More data insight

- Specific domain knowledge

4

## Deployment

- Implementation into production with a real-time analytics pipeline

# CONTACT


www.github.com/yaterjo

jly35630@yahoo.com

555-867-5309

Atlanta, Georgia

# Our Team

**Johnathan Yater**

*Data Science*

# THANK YOU