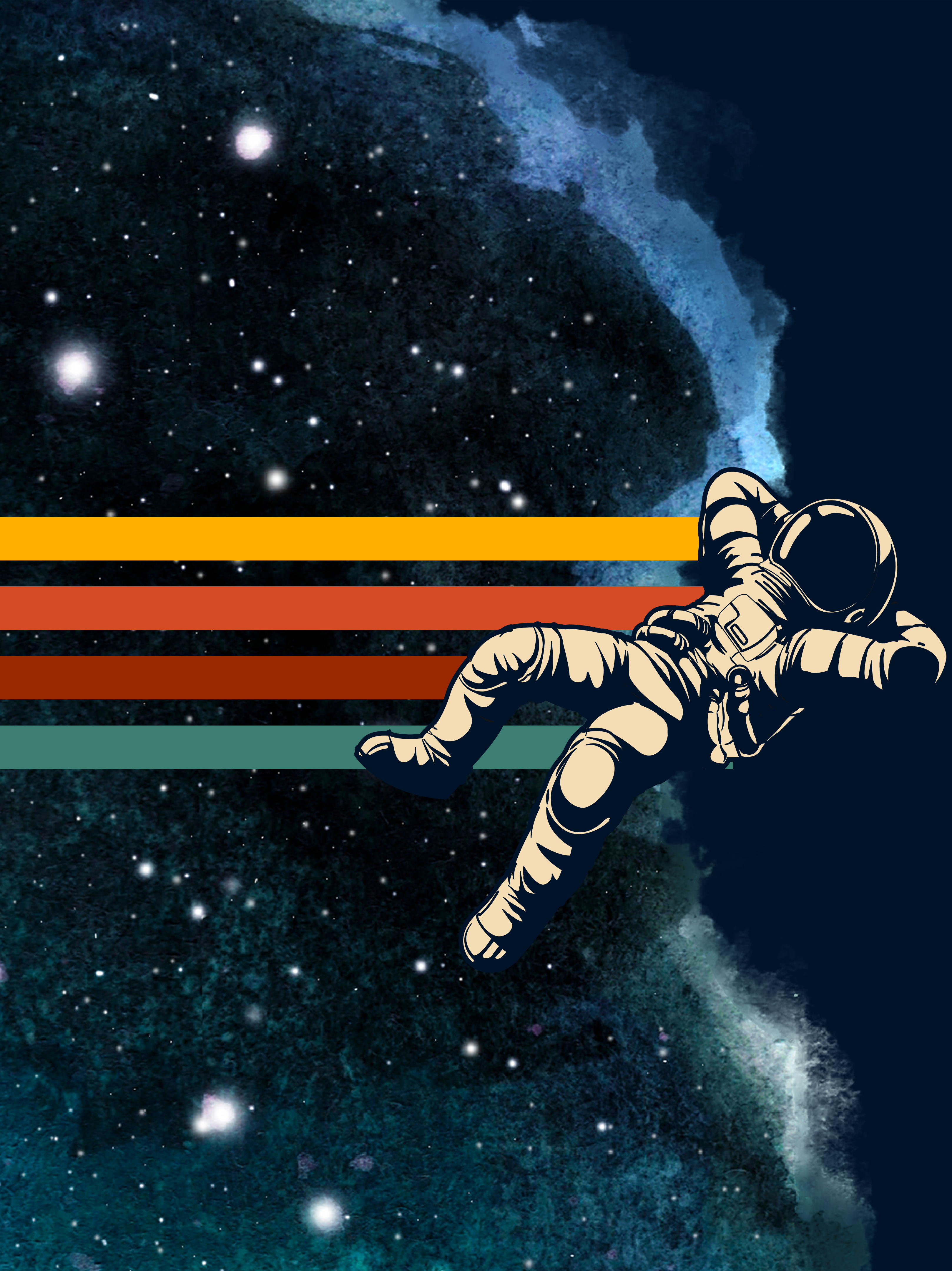

# NIMBLENESS COUNCIL

Flight Anomaly Detection and Prediction

# Executive Summary

This Slides presents a simplified Python program for predictive maintenance in aerospace systems, focusing on anomaly detection and prediction. The program utilizes machine learning techniques to analyze flight data and identify potential issues before they escalate. The methodology involves generating a mock dataset, training an Isolation Forest model, and evaluating its performance.

# Problem Statement

The aerospace industry faces challenges in ensuring the safety and reliability of flights. Unexpected failures in critical systems can lead to costly downtime and compromise safety. The predictive maintenance system aims to address these challenges by using AI to detect anomalies in flight data and predict potential issues.

# Methodology

## 01 Dataset Generation

A mock dataset was generated to simulate normal and anomalous flight data. Normal data was generated from a standard normal distribution, while anomalous data was generated from two additional distributions with different means.

## 02 Machine Learning Model

An Isolation Forest model was chosen for anomaly detection. The model was trained on a subset of the generated data, and predictions were evaluated on a test set. The contamination parameter was set to 0.05 to account for the expected proportion of anomalies.

## 03 Feature Engineering

Feature engineering is a crucial step in enhancing the model's ability to detect anomalies. Participants are encouraged to create new features from the existing data, such as aggregating sensor readings over time, calculating moving averages, or incorporating domain-specific knowledge.

# IMPLEMENTATION STRATEGY

## Prototype Development

Build a functional prototype to showcase the predictive maintenance capabilities.

## Testing and Refinement

Build a functional prototype to showcase the predictive maintenance capabilities.

## Full-Scale Deployment

Launch the predictive maintenance system for commercial use with continuous monitoring and updates.

# RESULTS

## Model Performance

The Isolation Forest model demonstrated promising results in detecting anomalies. The classification report showed good precision, recall, and F1 score for both normal and anomalous classes. The visualization highlighted the model's ability to distinguish between normal and anomalous data points.

# Limitations

## Mock Dataset Realism

The mock dataset may not fully capture the complexities of real flight data. Real-world anomalies could exhibit more nuanced patterns not present in the simulated dataset.

## Model Generalization

The model's performance on the simulated dataset does not guarantee its effectiveness on diverse real-world scenarios. External factors and variations in flight conditions need to be considered.

## Imbalanced Data

The imbalance between normal and anomalous instances in the dataset may impact the model's ability to generalize well to real-world scenarios with varying anomaly rates.

# Future Work

## Real Data Integration

Replace the mock dataset with real flight data to assess the model's performance under actual operating conditions and anomalies.

## Dynamic Thresholds

Explore adaptive thresholding techniques to dynamically adjust anomaly detection thresholds based on changing conditions or system behaviors.

## Multi-sensor Fusion

Investigate the integration of data from multiple sensors to enhance the model's robustness and provide a more comprehensive view of system health.

# INPUT

Dataset: The program takes a dataset as input, which includes historical flight data. This dataset consists of various parameters recorded during flights, such as altitude, airspeed, engine performance, and other relevant sensor readings. The dataset is expected to contain both normal and anomalous instances.

Feature Engineering (Optional): If participants choose to perform feature engineering, they can input the dataset into this process. Feature engineering involves creating new features from existing data to enhance the model's ability to detect anomalies. This could include aggregating sensor readings, calculating statistical measures, or incorporating domain-specific knowledge.

# PROCCES

Isolation Forest Model Training: The program splits the dataset into training and testing sets. An Isolation Forest model is then trained on the training set. The Isolation Forest algorithm is particularly chosen for its ability to efficiently isolate anomalies.

Anomaly Detection: The trained model is applied to the testing set for anomaly detection. Each data point is assigned an anomaly score based on its isolation in the decision tree structure of the Isolation Forest. A threshold is set to classify instances as either normal or anomalous.

# OUTPUT

Model Evaluation: The program evaluates the performance of the Isolation Forest model using metrics such as precision, recall, and F1 score. This evaluation provides insights into how well the model can distinguish between normal and anomalous instances.

Visualization: A scatter plot is generated to visually represent the model's predictions. Normal and anomalous data points are differentiated by color, and the model's predictions are overlaid on the plot. This visualization offers a clear illustration of the model's effectiveness.

# User Interaction

### Input Type

The program interacts with the user by asking for the type of input (e.g., simulated data, real flight data).

### Feature Engineering (Optional)

If participants choose to perform feature engineering, the program prompts them to input the dataset for this process.

### Output Type

The user specifies the desired output type, whether it's model evaluation metrics, a visualization, or both.

### Parameters

The user may also provide parameters such as the contamination rate (proportion of anomalies) for the Isolation Forest model.

# Revenue Model

## Subscription-based Model

Offer tiered subscription plans based on the size of the fleet and the frequency of predictive maintenance reports.

## One-time Implementation Fee

Charge an upfront fee for the integration and customization of the predictive maintenance system into the client's existing infrastructure.

# Conclusion & Recommendations

- The program concludes by summarizing the results, discussing limitations, and suggesting areas for future improvement. It emphasizes the need for real-world data integration, dynamic thresholding, multi-sensor fusion, online learning, and human-in-the-loop integration for a more robust predictive maintenance system in aerospace applications.

- In summary, the predictive maintenance program takes flight data as input, processes it through training an Isolation Forest model and performing anomaly detection, and provides outputs such as model evaluation metrics and visualizations. User interaction involves specifying input types, optional feature engineering, output preferences, and parameter settings.

# SPICEL THANKS TO THE NIMBLENESS COUNCIL

- Abdulaziz Jandaly — Project Manager
- Abdullah Alkathiri — Data Analyst , AI programmer
- Abdullah Alqarni — Front-end Developer
- Mohammed saeed — System Analyst
- Thamer Quaish — Programmer