# Detecting Anomalies in Satellite Data Using MindsDB

## Task 1: Pre-process Sensor Data

- **Description**: Clean the satellite sensor data and make it ready for anomaly detection.
- **Expected Output**: A pre-processed dataset that can be used by the machine learning model.

## Task 2: Detect Anomalies

- **Description**: Apply MindsDB's machine learning capabilities to detect anomalies such as temperature spikes or sensor failures in the satellite data.
- **Expected Output**: A list of detected anomalies, each flagged with severity and timestamp.

## Task 3: Report Generation

- **Description**: Generate a report summarizing the detected anomalies and their potential impact on satellite operations.
- **Expected Output**: A detailed PDF report with graphs and tables showing the anomalies.

## Task 4: Real-Time Alerts

- **Description**: Monitor satellite data continuously and trigger alerts when a significant anomaly is detected.
- **Expected Output**: Real-time notifications sent to the satellite operations team.

# Code Example: Anomaly Detection Crew

```
from crewai import Agent, Task, Crew, Process
import pandas as pd
import mindsdb  # Example MindsDB integration

# Define the agents
data_preprocessor = Agent(
    role='Data Preprocessor',
    goal='Clean and preprocess satellite sensor data for analysis',
    backstory='You clean and prepare raw sensor data for the machine learning model.',
    tools= ['pandas', 'numpy']
)

anomaly_detector = Agent(
    role='Anomaly Detector',
    goal='Detect anomalies in satellite data using MindsDB',
    backstory='You are a machine learning model specialized in identifying anomalies in sensor data.',
    tools=['MindsDB']
)

report_generator = Agent(
    role='Report Generator',
    goal='Compile detected anomalies into a detailed report and notify stakeholders',
    backstory='You summarize the findings from the anomaly detection and notify relevant teams.',
    tools=['matplotlib', 'Slack', 'Email', 'GPT']
)

# Define the tasks
```

```
data_preprocessing_task = Task(
    description='Clean and preprocess satellite sensor data',
    expected_output='A preprocessed dataset ready for anomaly detection',
    agent=data_preprocessor
)

anomaly_detection_task = Task(
    description='Use MindsDB to detect anomalies in satellite sensor data',
    expected_output='A list of anomalies detected in the data',
    agent=anomaly_detector
)

report_generation_task = Task(
    description='Generate a report of detected anomalies and notify stakeholders',
    expected_output='A detailed PDF report with anomaly data and visualizations',
    agent=report_generator
)

# Create the crew
crew = Crew(
    agents= [data_preprocessor, anomaly_detector, report_generator],
    tasks= [data_preprocessing_task, anomaly_detection_task, report_generation_task],
    process=Process.sequential
)

# Kickoff the process
result = crew.kickoff(inputs= {'satellite data': 'path/to/sensor_data.csv'})
print(result)
```