# Rescue Decision Systems – ReadMe

## Overview

Rescue Decision Systems (RDS) is a machine-learning and GIS-driven platform designed to assist search and rescue (SAR) coordinators in evaluating SARSAT distress alerts. The system integrates historical data, real-time weather, and vessel tracking to improve decision-making and reduce false alarms.

## Key Features

- GIS Mapping & Weather Analysis: Visualizes SARSAT alerts, interpolated weather data, and satellite ground tracks.

- Machine Learning (ML) Alert Analysis: Predicts the probability of an alert being an actual distress case.

- Vessel & Aircraft Tracking: Assesses distress scenarios based on ship routes, time at sea, and environmental factors.

- International Collaboration: Enables SAR teams to share active cases securely.

- User Authentication & Access Control: Restricts access to authorized users based on permissions.

- Dual Alert Location Handling: Supports analysis and visualization of both 'A' and 'B' locations for alerts, ensuring coverage for multiple RCC areas of responsibility (AORs).

- Range Ring Visualization: Uses position resolution (accuracy) to generate range rings around alert locations for better spatial awareness.

- Consistent Measurement Units: Displays distances in meters and feet, and temperatures in Fahrenheit and Celsius for user clarity.

## Development Roadmap

### Phase 1: GIS & Weather Integration

- Develop local GIS visualization of alerts.

- Incorporate real-time weather data.

- Deploy to DigitalOcean.

### Phase 2: Machine Learning Implementation

- Train an initial ML model for distress likelihood prediction.

- Store structured SARSAT alert data in SQL.

- Deploy updates to the server.

### Phase 3: Distress Scenario Modeling

- Integrate vessel and aircraft tracking.

- Estimate weather impact along the distress route.

- Improve ML predictions based on travel patterns.

### Phase 4: Advanced Weather Data & Web Scraping

- Fetch high-resolution forecasts from external sources.

- Optimize predictive analysis.

- Deploy additional features.

### Phase 5: Expansion to Canada & Mexico

- Adapt system for cross-border SAR operations.

- Ensure compatibility with RCC message formats.

- Deploy updates for international users.

## Installation & Setup

### Requirements

- Python 3.8+

- PostgreSQL 13+

- Flask & Flask-Login

- Pandas, NumPy, SciKit-Learn

- DigitalOcean (or similar VPS for hosting)

### Local Setup

Clone the repository and install dependencies:

```bash  
git clone https://github.com/your-repo/RDSwebsite.git  
cd RDSwebsite  
pip install -e .  
flask run  
```

## Deployment

Push changes to GitHub.

SSH into the DigitalOcean droplet:

```bash  
ssh user@server-ip  
cd /path/to/app  
git pull origin main  
systemctl restart gunicorn  
```

Verify web interface is updated.

## Contributing

We welcome contributions! To get involved:  
1. Fork the repository.  
2. Create a feature branch.  
3. Submit a pull request with clear documentation.

## License

MIT License. See `LICENSE.md` for details.