SeismoScan

Developing Machine Learning Tools to Map Active Faults in Bathymetry Data

Team:
Noah Carges - Thomas Rotchford,
Alyssa Sombrero - Brady Wisniewski

Mentor:
Scott LaRocca

NAU NORTHERN ARIZONA UNIVERSITY

Our Clients - NAU School of Earth and Sustainability

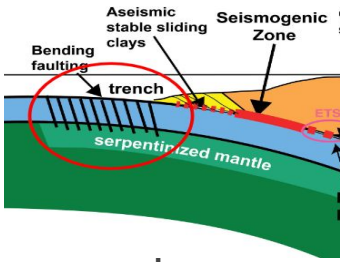
- They are Professional Geologists and Seismologists working at the NAU School of Earth and Sustainability
- As part of their work they study the impact of faults and how they impact important geological processes

Dr. Donna Shillington
Professor of Geology

Dr. James Gaherty
Professor of Geology

Dr. Christine Regalla
Associate Professor of Geology

The Problem



- Our Clients study processes such as earthquakes and the deep water cycles, which currently requires faults to be hand mapped.
  - An in house effort took around **80** hours to map around **40,000** square kilometers
  - Some estimates have the total size of fault zones reaching 5 million square kilometers
- Even by hand mapping, there's always a consistent problem of:
  - Miscalculations: With manual calculations, there is always a human error
  - Missing faults: Researchers focus on larger faults for too long, they miss other faults
  - Inconsistent Results: Introduces variations and errors from fault interpretations
- These factors are important to fix in order gain reliant and efficient data for important research

Key Requirements

Must Haves

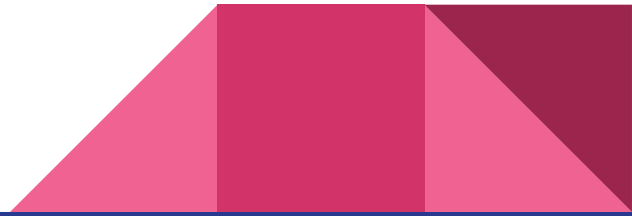
- Machine Learning Model
- Command Line Interface
- Output points of faults to Text File
- Compatible with MacOS, Linux, and Windows

Nice to Have

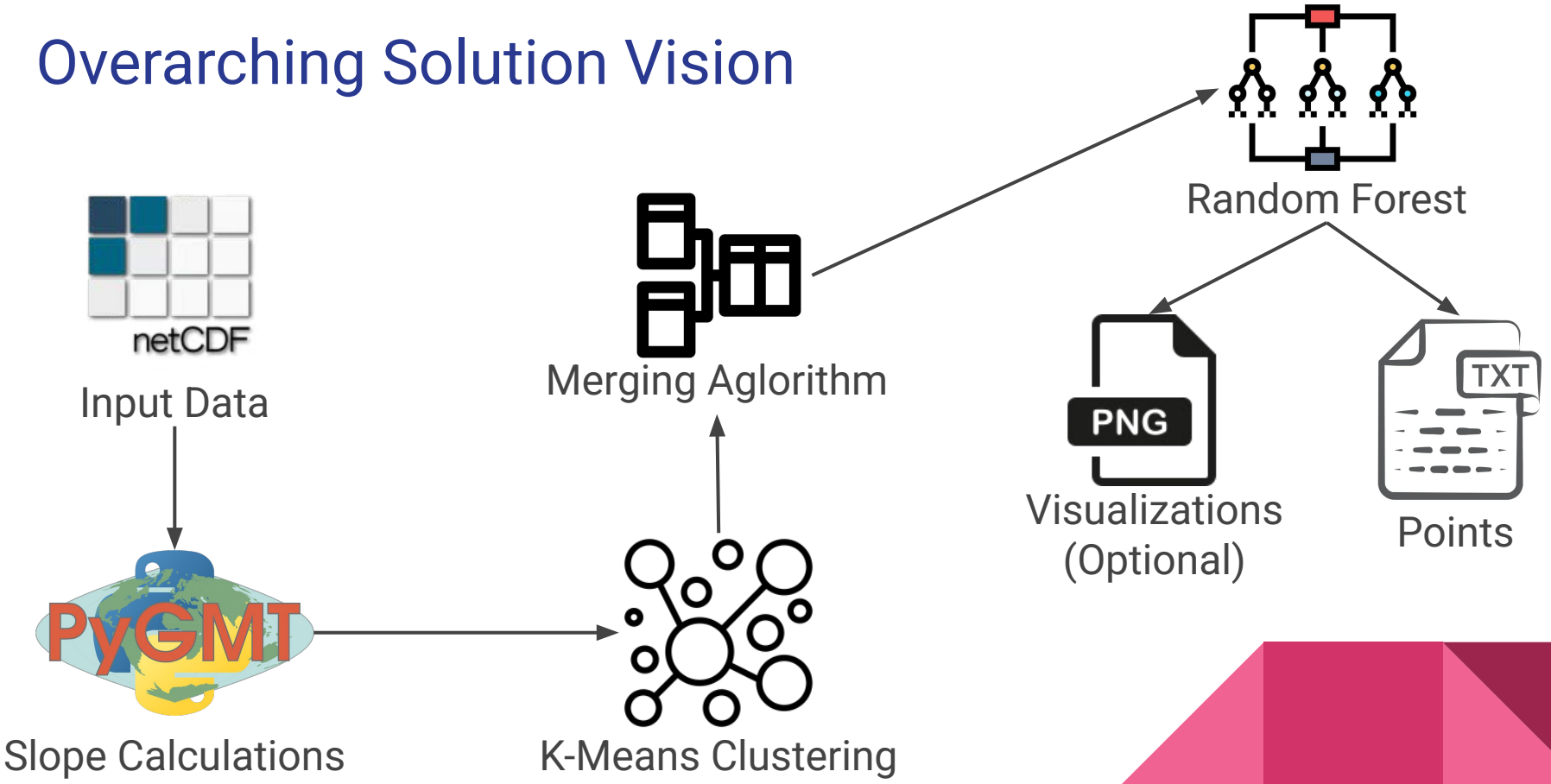
- Configuration management
- Visualization of Ouptuts
- Log File
- Optional progress display
- File/model input manager
- Model confidence statistics

Won’t Have

- User Interface/User Experience
- Database integration



Overarching Solution Vision



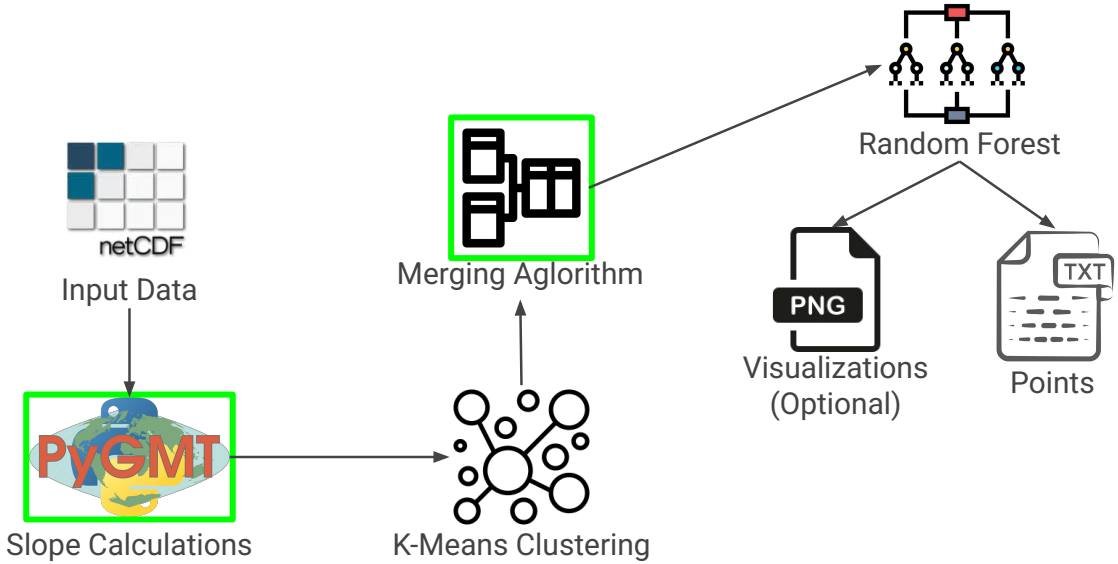
Final Solution Overview - Feature Engineering

Slope Calculation:

- First Derivative
- Configurable number of angles
- Unit Independant
- NaN handling

Merging:

- Raster-space grouping
- Depth merging
- Directional merging
- Bridging neighbors



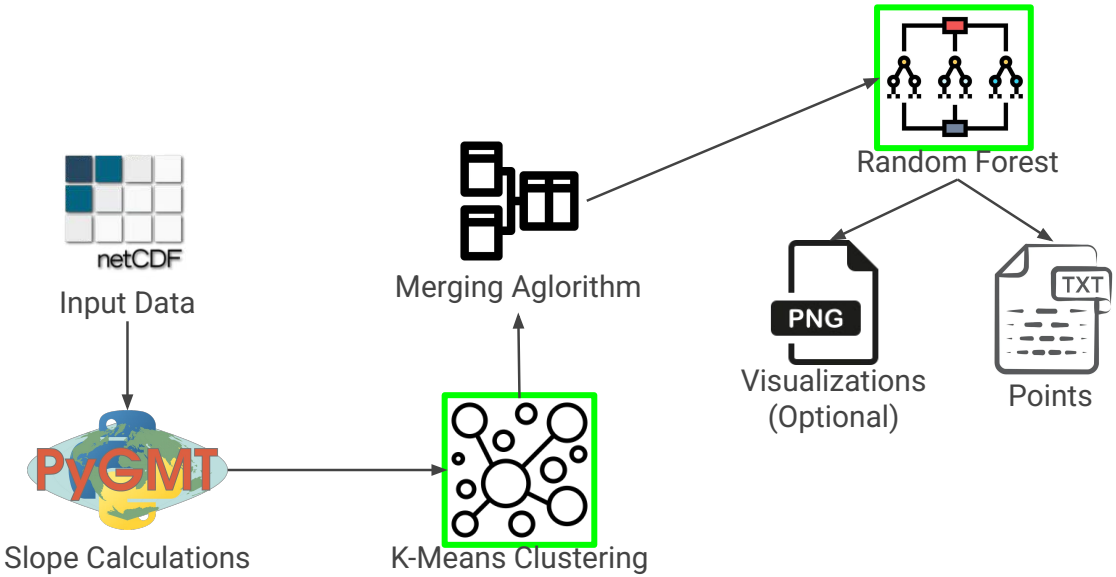
Final Solution Overview - ML Pipeline

Clustering:

- Takes in all calculated slopes
- Two clusters (fault/no-fault)
- First stage of fault detection
- Chunk by chunk process

Random Forest:

- Pre-trained object classification models
- Evaluates features (shape, linearity, etc)
- Object → Probability
- Probability threshold (adjustable)



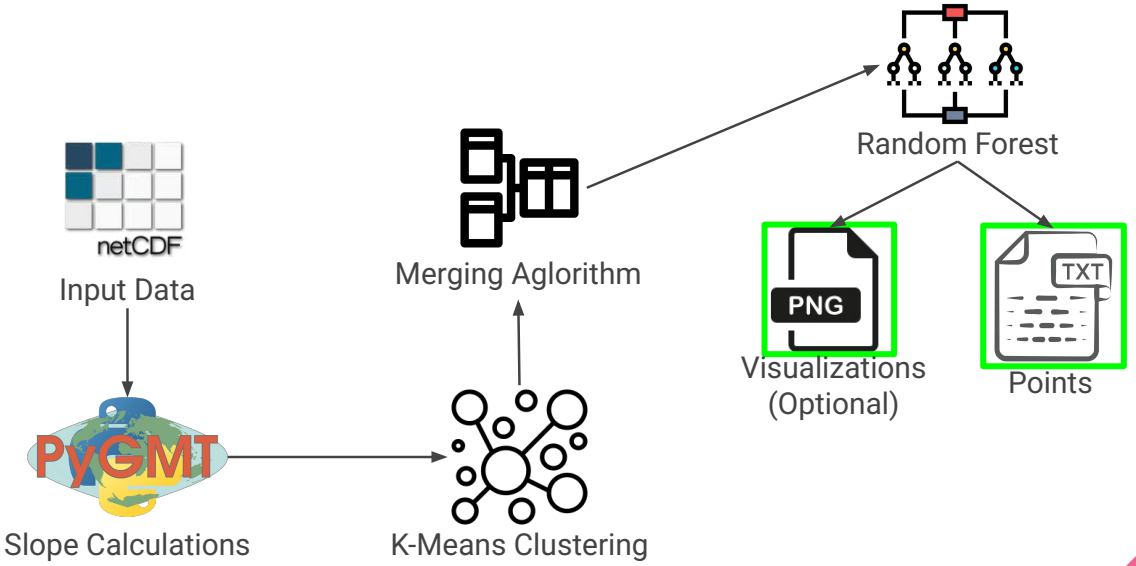
Final Solution Overview - Outputs

Text Outputs:

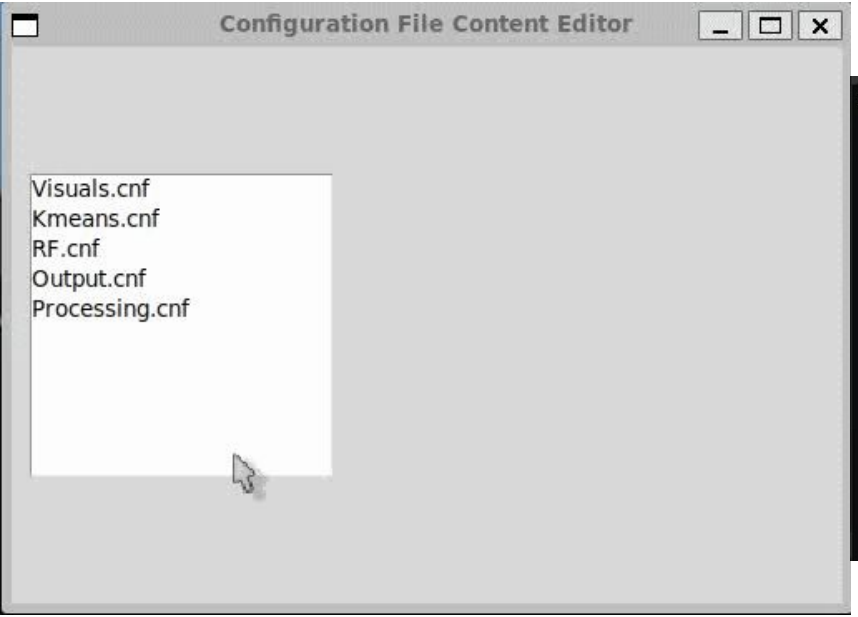
- K-Means
- Merged
- Random Forest
- Delimiter Seperated

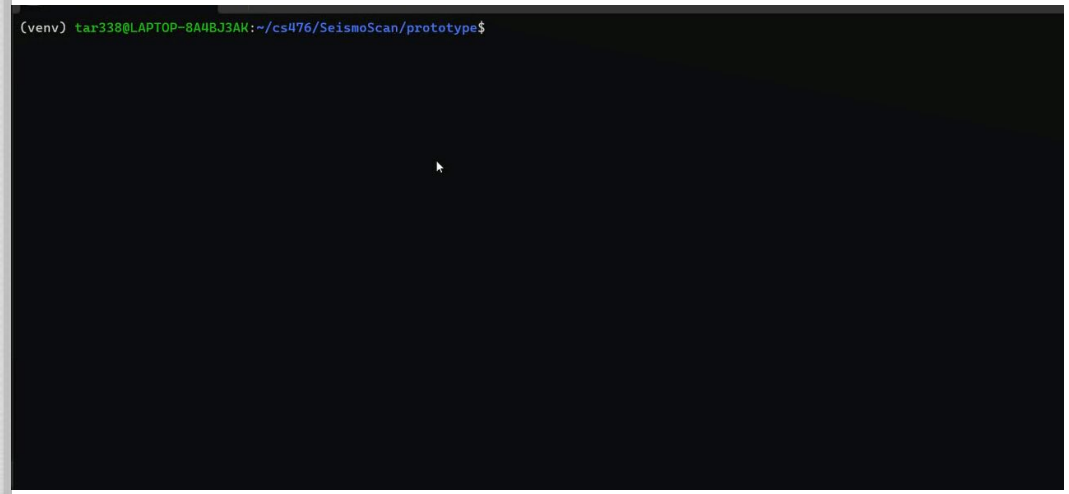
Visuals:

- Gradient
- K-Means
- Merged
- Random Forest

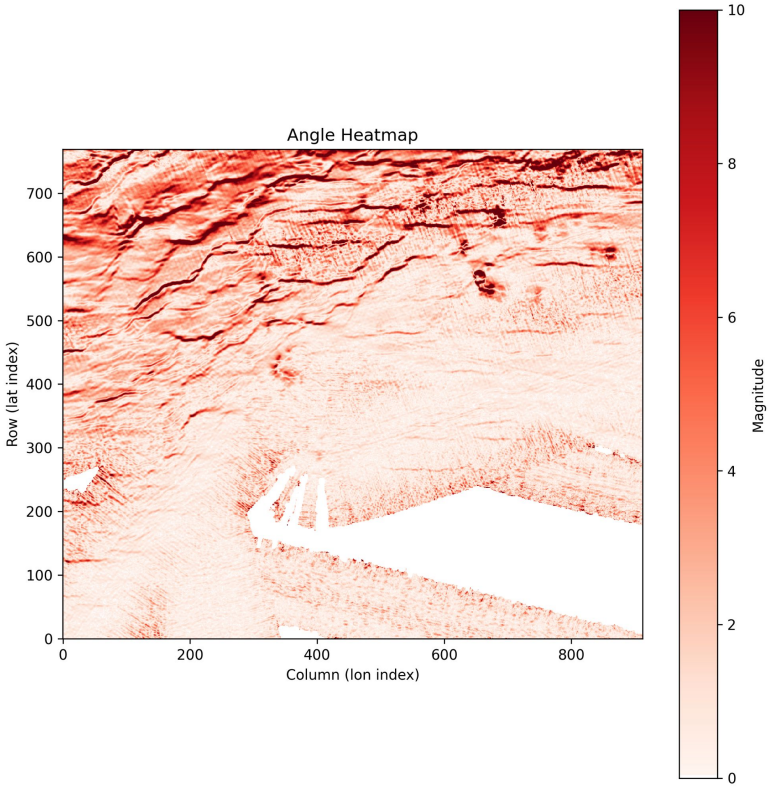


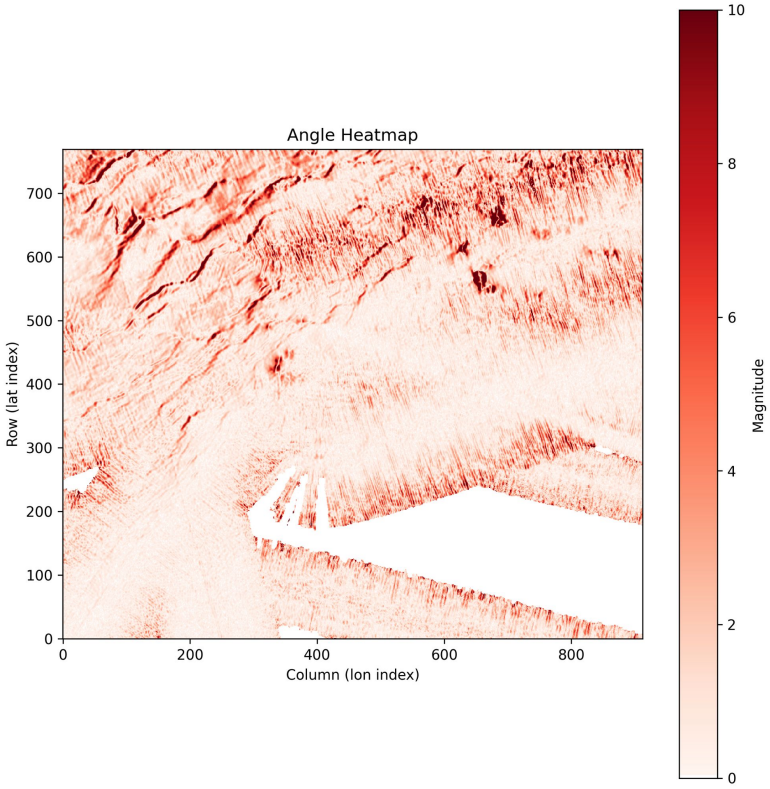
Product Demonstration



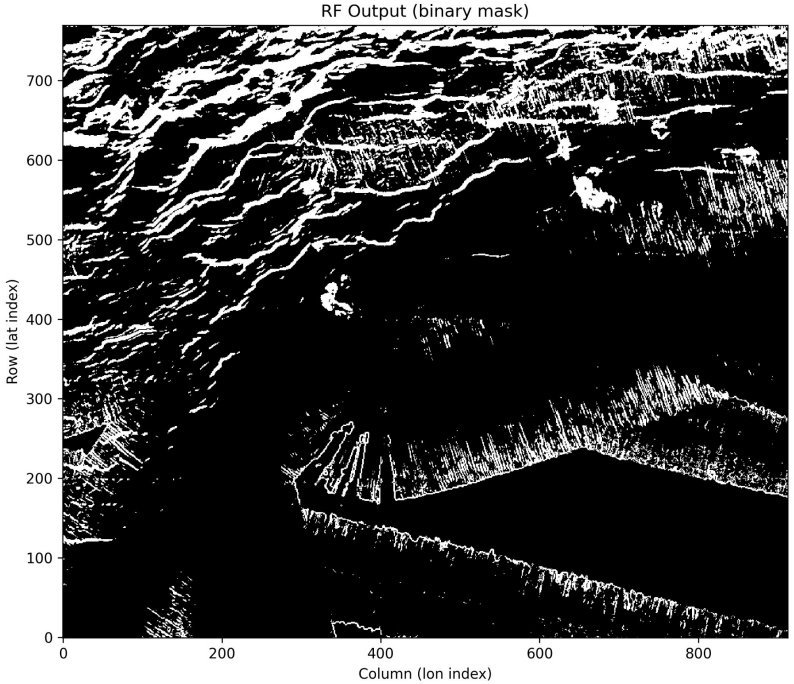


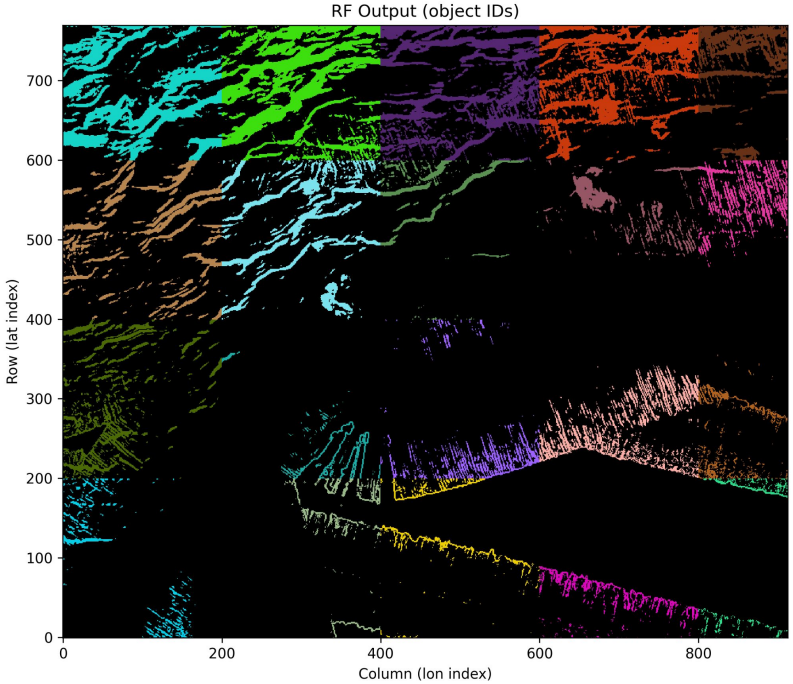
Gradient


Angle Heatmap
Row (lat index)
Column (lon index)
Magnitude
0
100
200
300
400
500
600
700
0
200
400
600
800
0
2
4
6
8
10

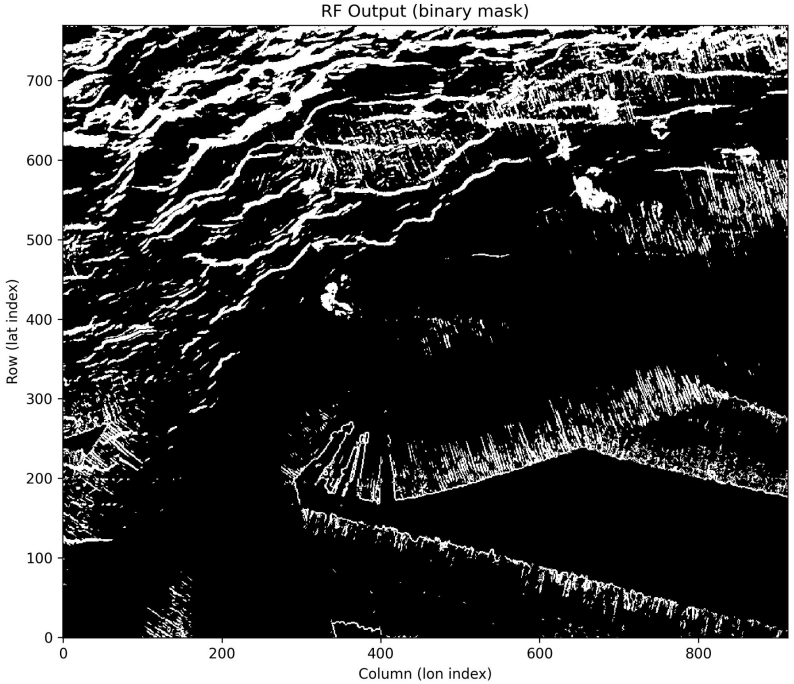

Angle Heatmap
Row (lat index)
Column (lon index)
Magnitude
0
100
200
300
400
500
600
700
0
200
400
600
800
0
2
4
6
8
10

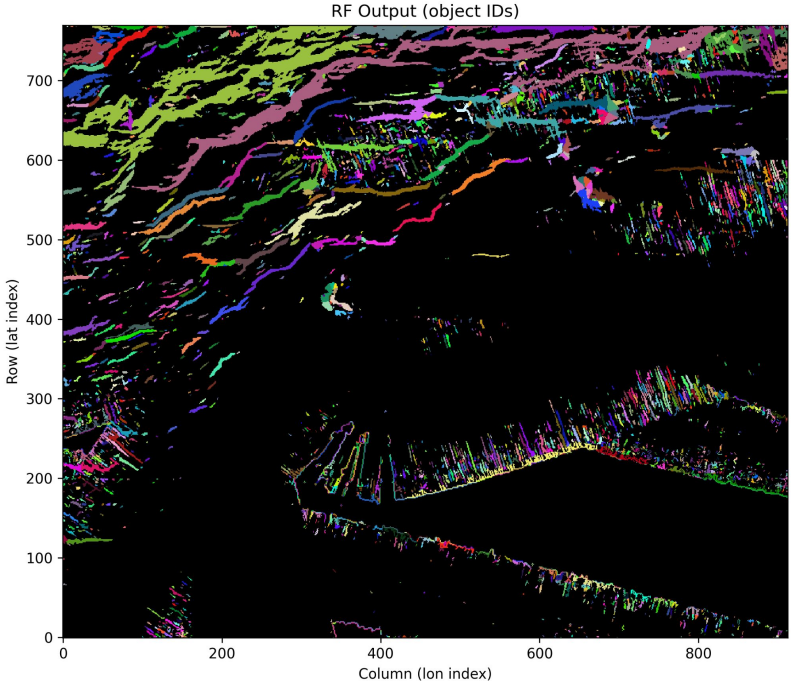
K-Means


RF Output (binary mask)
Row (lat index)
Column (lon index)
0
100
200
300
400
500
600
700
0
200
400
600
800

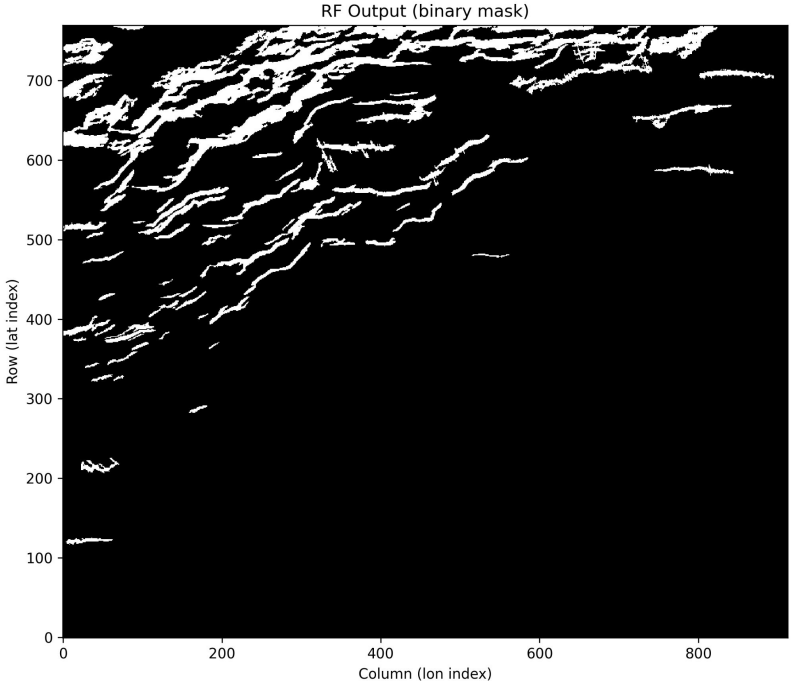

RF Output (object IDs)
Row (lat index)
Column (lon index)
0
100
200
300
400
500
600
700
0
200
400
600
800

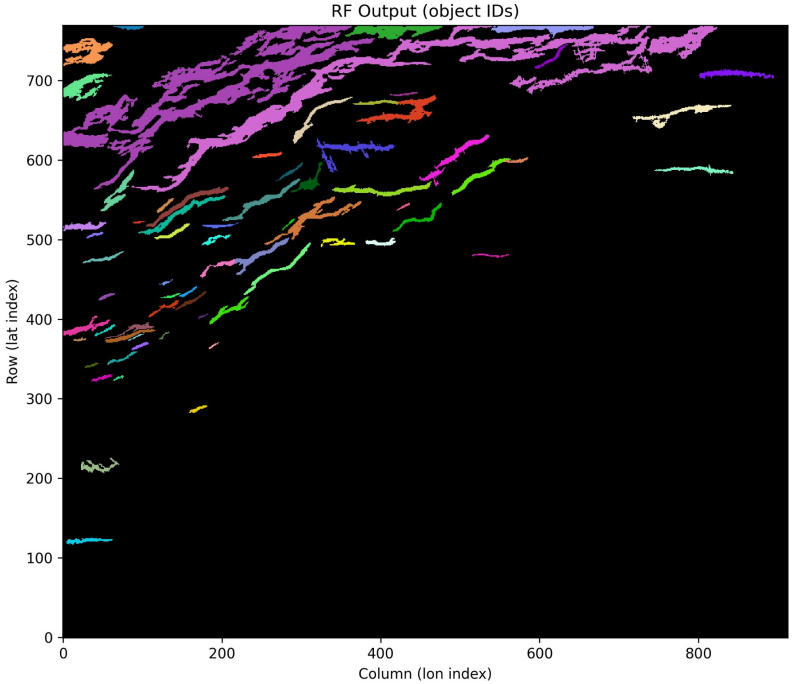
Merged


RF Output (binary mask)
Row (lat index)
Column (lon index)
0
100
200
300
400
500
600
700
0
200
400
600
800


RF Output (object IDs)
Row (lat index)
Column (lon index)
0
100
200
300
400
500
600
700
0
200
400
600
800

Random Forest





Testing - Accuracy & Efficiency Tests

Accuracy Testing:

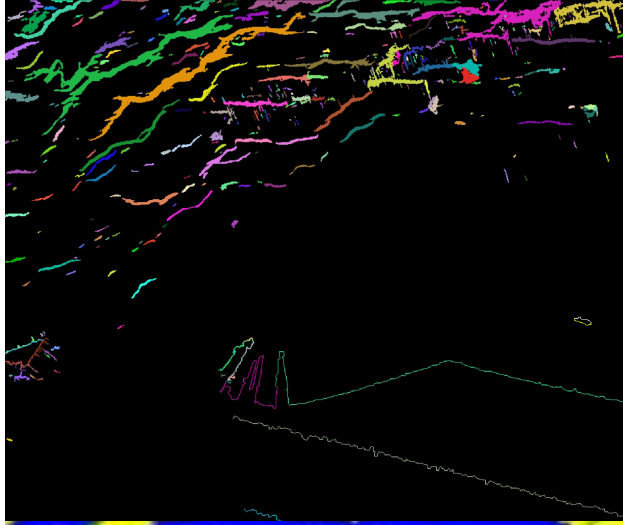
Model outputs include a confusion matrix calculated on the hand interpreted faults we have retrieved from our clients

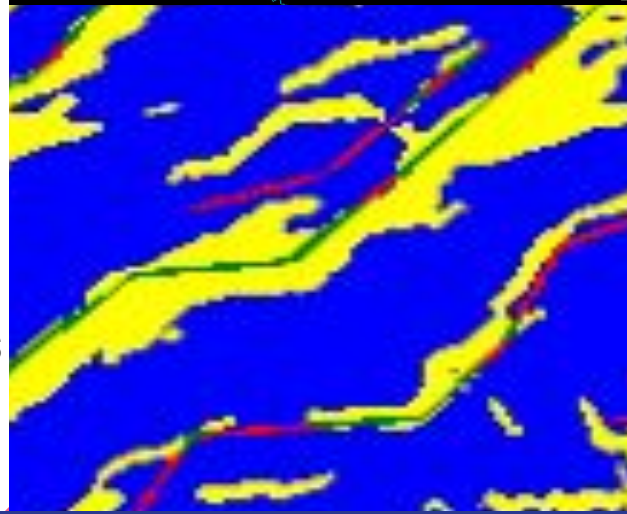
- True positives/false negatives
- We used this to select the best performing models

Efficiency Testing:

Recording how model runtimes scale with data size

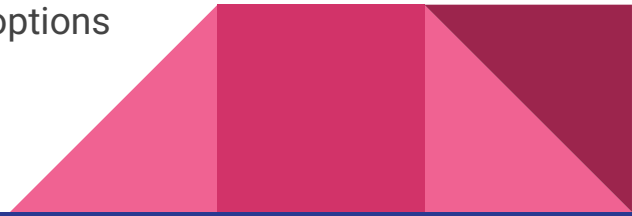
- Calculations for runtimes on different sized datasets
- We prioritized accuracy over efficiency



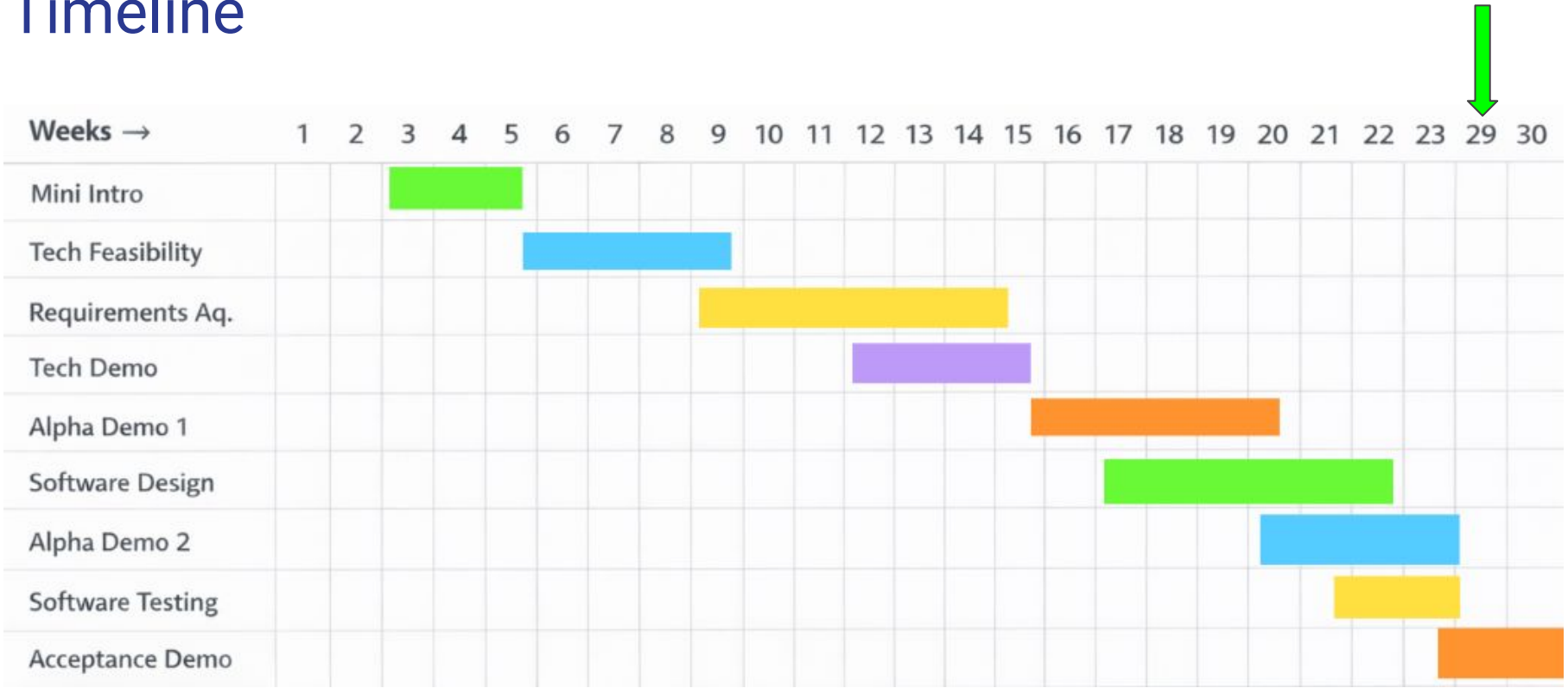


Challenges & Resolutions

- **Machine Learning Knowledge**
  - Trial and error along with structured separation of tasks
- **Pipeline Decision**
  - Planning over one step/phase, then discussions on how to use that output as an input later
- **Feature Engineering**
  - Each phase designed to detect aspects of faults, features build to to compliment those goals
- **Input Data**
  - Requirements over specific structure of filetype, resolution, missing data handling
- **Regional Generalizability**
  - Continuous testing of different configurations of all analysis options



Timeline


Weeks →
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 29 30
Mini Intro
Tech Feasibility
Requirements Aq.
Tech Demo
Alpha Demo 1
Software Design
Alpha Demo 2
Software Testing
Acceptance Demo

What Could Come Next?

Explore Even More Models
Test other supervised and unsupervised models on the dataset to compare accuracy performance

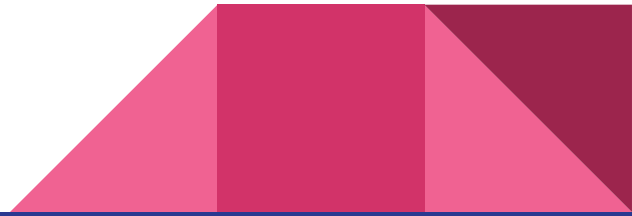
Optimize for Specific Areas
Spending time testing and training specific models that would be better at recognizing features in other parts of the world

User Friendliness
Add options to improve user experience such as saving preferred configurations, and queuing up multiple grids, models, or configurations

Expanding Upon Feautres
Add options to search for and label more features than just faults, including things like Sea Stacks, Volcanoes, Trenches, etc

Conclusion

- We have worked alongside our clients to decide:
  - Requirements
  - Structure
  - Process
- We have provided comprehensive options to allow users to tune SeismoScan to whatever their needs may be
- We have provided our clients with a viable product that will help geologists at NAU map faults



Any Questions?