Predicting Expected Travel Time (ETT) Using AIS Data

Software Development Project, SS2025
Group 4 - Kateryna Makarova

Technische Universität Hamburg (TUHH)
Supervisor: Dipl. Informatiker Rainer Marrone

TUHH

1.Problem Decription

- AIS data is noisy, incomplete, and high-volume.
- Ports need accurate ETT to optimize berth planning and operations.

Goal: Predict ETT for a vessel based on real-world AIS input.

2. 1. What We Have Built

✅ Developed a Streamlit web app for predicting vessel ETT using AIS data, incl. Docker

✅ Performed AIS data cleaning & feature selection

✅ Built and compared 4 machine learning models:

- Linear Regression
- Random Forest
- MLP Regressor
- Gradient Boosting

✅ Implemented a broker agent to combine model outputs

✅ Designed interactive visualizations:

- Upload coordinates
- Map with historical & predicted routes
- ETA popups + downloadable results

✅ Added error analysis: metrics, heatmaps, top outliers

✅ Fully documented: scripts, video instruction, README.md

2.2. What Is Still Missing

⚠️ No live AIS data integration. Current implementation uses static test data, file upload available

⚠️ Incomplete test coverage: only detection and removal of NaNs, handling of missing/invalid values, checks for values out of expected bounds, verification when no file is uploaded, protection against invalid user inputs

⚠️ GUI could include advanced analytics (e.g., congestion, delays)

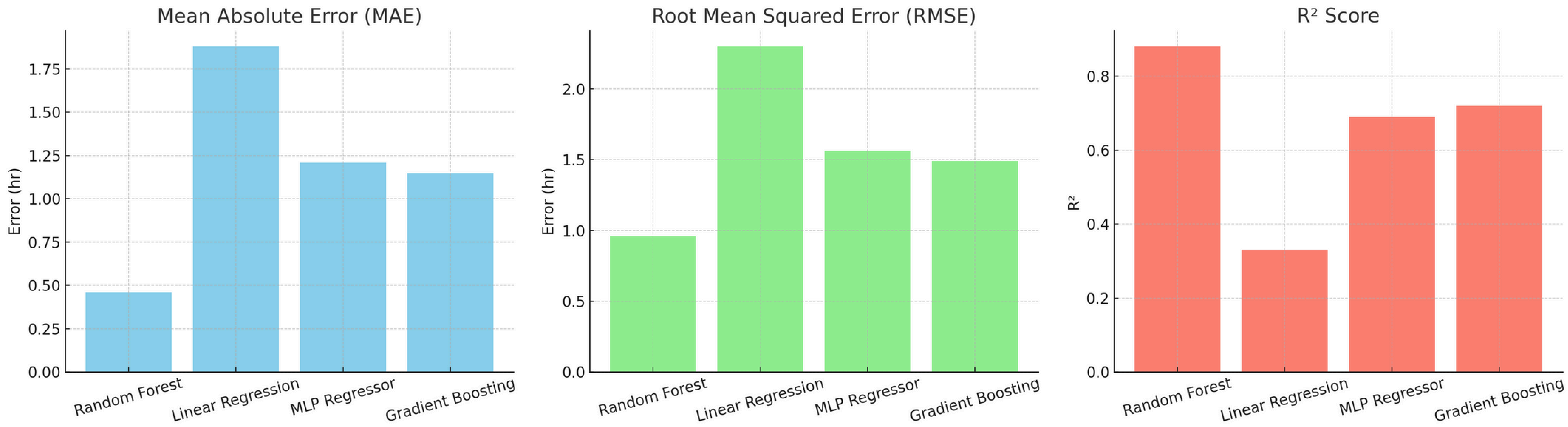
⚠️ Broker Agent optimisation

3.1. Machine Learning Models

Model	MAE (hr)	RMSE (hr)	R^2
	Mean Absolute Error – average difference between predicted and actual ETT.	Root Mean Squared Error – like MAE but penalizes large errors more heavily.	R-squared – how well the model explains variation in ETT (1.0 = perfect).
Random Forest	0,46	0,96	0,88
Linear Regression	1,88	2,3	0,33
MLP Regressor	1,21	1,56	0,69
Gradient Boosting	1,15	1,49	0,72

3.2. Machine Learning Models

Visual representation of Metrics



Random Forest demonstrates the best results so far

3.2. Machine Learning Models (with Tuning)

Model	***Tuned Parameters***	***Reason for Parameter Choice***	***Tuning Method***	***Observed Impact***	***Metrics After Tuning***
Random Forest	`n_estimators`, `max_depth`, `min_samples_split`	• More trees reduce variance • Depth limits overfitting • Split size balances bias/variance	RandomizedSearchCV+ 3-fold CV (MAE)	Small improvement — already strong base performance. Fine-tuned for robustness.	MAE - 0.78 RMSE - 1.16 R2 - 0.83
Gradient Boosting	`n_estimators`, `learning_rate`, `max_depth`	• More stages improve fit • Learning rate stabilizes updates • Depth limits complexity	RandomizedSearchCV+ 3-fold CV (MAE)	Moderate improvement in MAE and R^2 — better trade-off than base model.	"MAE": MAE - 0.22 RMSE - 0.31 R2 - 0.90
MLP Regressor	`hidden_layer_sizes`, `activation`, `alpha`	• Hidden layers define capacity • Activation affects learning • Alpha reduces overfitting	GridSearchCV + 3-fold CV (MAE)	Mixed results — slight improvement in accuracy, but training time and complexity increased.	MAE - 1.09 RMSE - 1.46 R2 - 0.73

4. Direct comparison of Tuned and Not Tuned Models

Before

Model	MAE (hr)	RMSE (hr)	R^2
	Mean Absolute Error – average difference between predicted and actual ETT.	Root Mean Squared Error – like MAE but penalizes large errors more heavily.	R-squared – how well the model explains variation in ETT (1.0 = perfect).
Random Forest	0,46	0,96	0,88
Gradient Boosting	1,15	1,49	0,72
MLP Regressor	1,21	1,56	0,69

After Tuning

Model	MAE (hr)	RMSE (hr)	R^2
	Mean Absolute Error – average difference between predicted and actual ETT.	Root Mean Squared Error – like MAE but penalizes large errors more heavily.	R-squared – how well the model explains variation in ETT (1.0 = perfect).
Random Forest	↑ 0,78	↑ 1,16	↓ 0,83
Gradient Boosting	↓ 0,22	↓ 0,31	↑ 0,9
MLP Regressor	↓ 1,09	↓ 1,46	↑ 0,73

Tuning improved models slightly, with the most significant gains observed in Gradient Boosting

5. Prediction Error Analysis – Tuned Gradient Boosting

Summary Statistics

Mean Error	0.0011 hr	near-zero bias
Median Error	0.0010 hr	highly symmetric predictions
75th Percentile	0.1552 hr	75% of predictions within ~10 minutes
Max/Min Error	+2.06 hr / –1.78 hr	

Error diagnostics were integrated directly into the Streamlit app, making them accessible for users and developers.

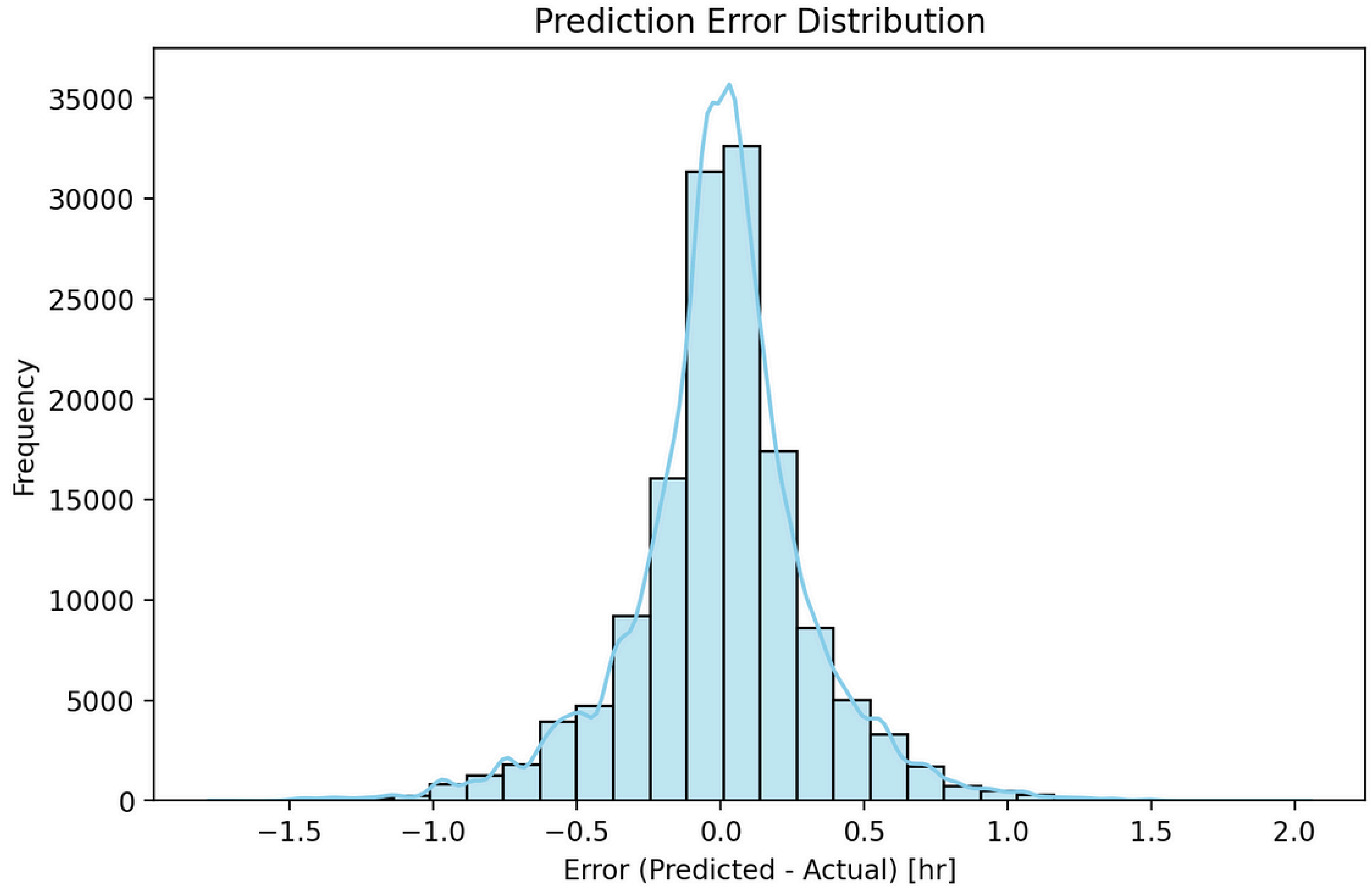
5.1. Prediction Error Analysis – Tuned Gradient Boosting

Prediction Error Distribution

✅ Tight bell-shaped histogram centered around 0

✅ Indicates low variance and minimal systematic bias

✅ No extreme outliers — model is stable



5.2. Prediction Error Analysis – Tuned Gradient Boosting

Spatial Error Mapping

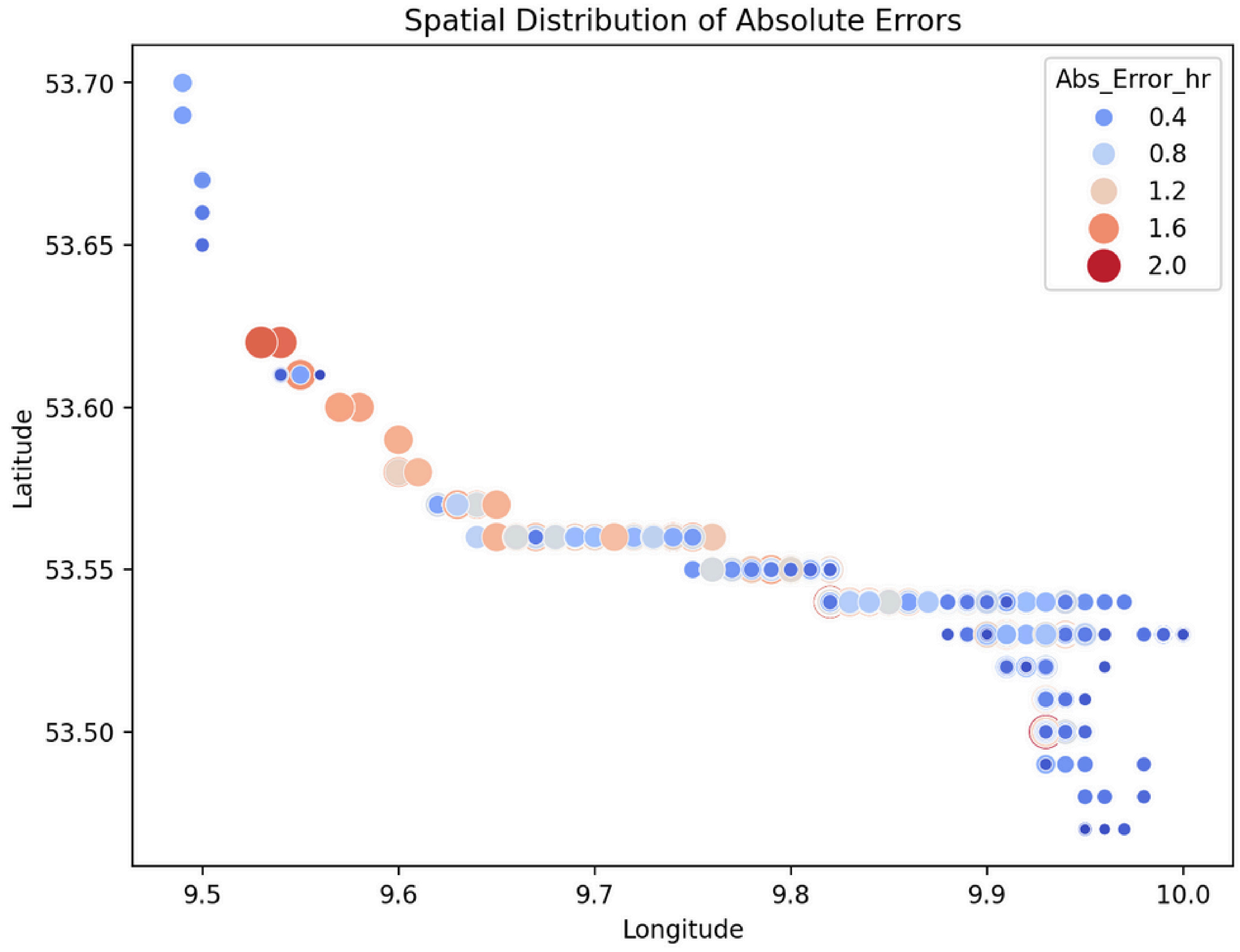
✅ Blue = low error, Red = higher error

✅ Most accurate predictions occur mid-journey

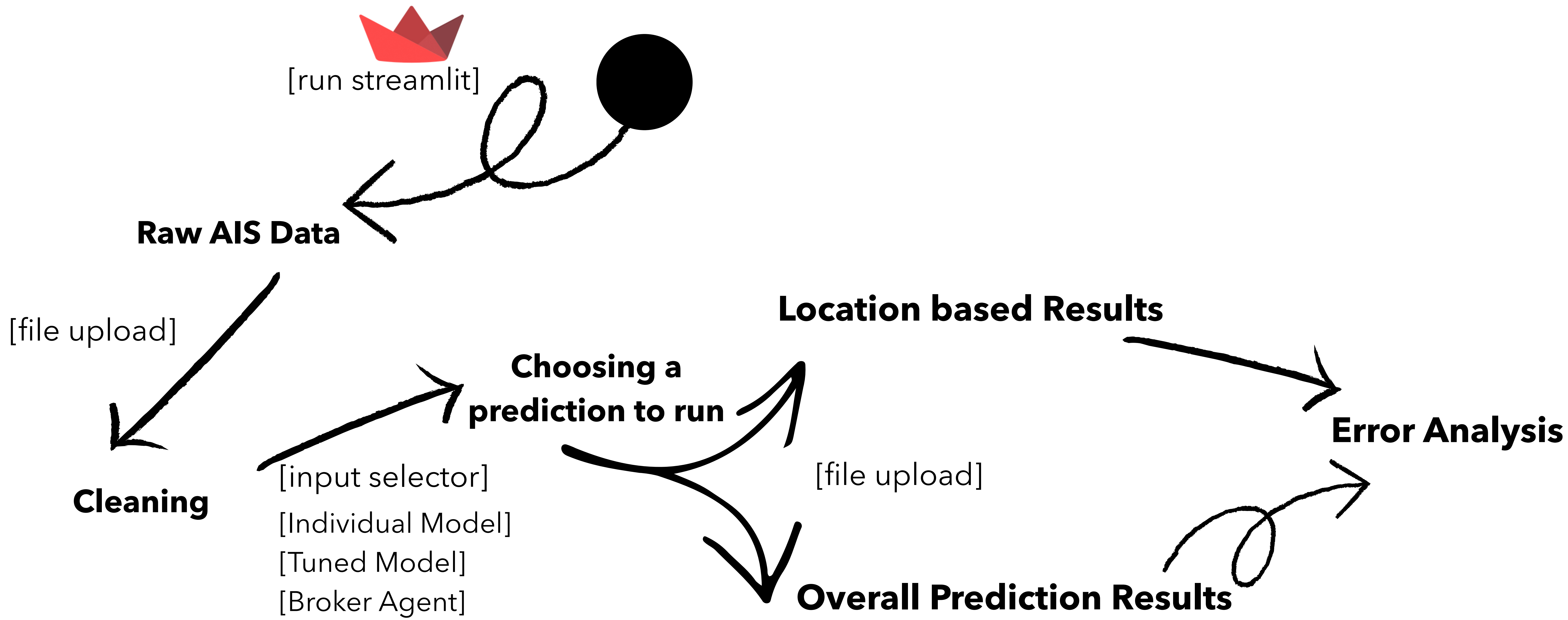
✅ Higher errors in harbor areas — likely due to unpredictable maneuvers or low-speed zones

🗺️ Hamburg Port
Latitude: 53.5461° N
Longitude: 9.9661° E

🗺️ Bremerhaven Port
Latitude: 53.5396° N
Longitude: 8.5809° E



6. Overall Architecture - Streamlit App, Docker



Entire pipeline containerized with Docker for reproducibility and deployment

7. Live Demo

🚢 Ship Trip Duration Prediction App

Upload your CSV file for prediction

Drag and drop file here
Limit 200MB per file • CSV

Browse files

Sample Data

	TripID	MMSI	StartLatitude	StartLongitude	StartTime	EndLatitude	EndLongitude	EndTime	time	shiptype	Length	Breadth	Draught	Latitude	Longitude	SOG	COG	TH	SOG_kmh	Distance_km	Bearing
0	10257	305506000	53.58	8.15	2016-02-26 21:04:00	53.53	9.91	2016-02-27 06:41:00	2016-02-26 21:04:00	71	161	25	8.71	53.58	8.15	2.5	40.5	34	4.63	116.3875	92.0298
1	10257	305506000	53.58	8.15	2016-02-26 21:04:00	53.53	9.91	2016-02-27 06:41:00	2016-02-26 21:04:00	71	161	25	8.71	53.58	8.15	2.6	56.6	43	4.8152	116.3875	92.0298
2	10257	305506000	53.58	8.15	2016-02-26 21:04:00	53.53	9.91	2016-02-27 06:41:00	2016-02-26 21:05:00	71	161	25	8.71	53.58	8.15	3	37.3	20	5.556	116.3875	92.0298
3	10257	305506000	53.58	8.15	2016-02-26 21:04:00	53.53	9.91	2016-02-27 06:41:00	2016-02-26 21:07:00	71	161	25	8.71	53.59	8.16	4.5	5.2	354	8.334	116.3875	92.0298
4	10257	305506000	53.58	8.15	2016-02-26 21:04:00	53.53	9.91	2016-02-27 06:41:00	2016-02-26 21:07:00	71	161	25	8.71	53.59	8.16	4	10.6	4	7.408	116.3875	92.0298

Choose prediction mode

- (●) Individual Model
- ( ) Broker Agent

Choose a model:

Gradient Boosting

Model version:

- ( ) Initial Model
- (●) Tuned Model

☐ Force model re-run

Run Individual Model

8. Conclusion & Outlook

We developed a robust Streamlit-based, Docker supported tool for predicting Expected Travel Time (ETT) using AIS data.

Our pipeline includes:

✅ Rigorous data cleaning
✅ Multiple ML models with and without hyperparameter tuning
✅ A broker-based decision architecture
✅ Error diagnostics and visualizations

Results:

✅ Gradient Boosting (tuned) showed the best performance (MAE ≈ 0.22 h).
⚠️ Errors are more frequent in harbor zones due to complex navigation behavior.

9. What's Next

- Integrate real-time AIS feeds for live prediction
- Improve error handling in port areas using domain-specific features
- Expand prediction to multi-stop or multi-trip vessel trajectories
- Deploy on a cloud server with API access for port authorities