Module 3 - ML
Analysis of delays in clinical trials

Daniëlle Verschoor
CAS Module 3

Overview

- Background
- Methods
- Analysis
  - Logistic regression
  - Random forest
  - XGBoost
  - Clustering
- Conclusion

Background – Clinical trials

- Clinical trials are *a type of research that studies new tests and treatments* and evaluates their effects on human health outcomes.
  - Test safety and efficacy of new, or old, drugs or medical devices
  - Very controlled by ethical regulations
- Key Phases:
  - Phase 1: Small group – test safety and dosage
  - Phase 2: Larger group – test effectiveness and side effects
  - Phase 3: Even larger – compare with current standard treatment
  - Phase 4: After approval – monitor long-term safety

Background - Registries

- Clinical trial registries are public databases with trial information
  - Study purpose, design, population specifics, location, status….
  - Promote transparency, accountability and trust
- Many different registries
  - 18 major ones, covering large regions/countries
    - Clinicaltrials.gov → USA
    - EU Clinical trial register → European Union/European Economic Area

Background – Clinicaltrial.gov

- Most used; even in European studies

Year	Event / Milestone	Description
1997	**Launch**	Established by the **U.S. National Library of Medicine (NLM)** and the **National Institutes of Health (NIH)** to provide public access to information about clinical trials.
2000	**Public availability**	The website went live, allowing researchers and the public to **search for trials**.
2005	**FDAAA 801 law**	The **Food and Drug Administration Amendments Act** required **registration and results reporting** for certain trials, increasing transparency.
2007	**Results database added**	Allowed researchers to submit **summary results** of clinical trials, including safety and efficacy outcomes.
2017	**Final rule implemented**	Clarified registration and reporting requirements under **FDAAA 801**, expanding compliance and penalties for missing data.
2020s	**Ongoing updates**	The registry continues to **expand coverage globally**, improve **data quality**, and provide **tools for analysis** of clinical trial data.

Aim ADICT project

- Many trials face a delay
  - For trialists common knowledge
  - Reasons are unclear

- ADICT aims to **A**nalysis of **D**elays **I**n **C**linical **T**rials
- Investigate if there are patterns in the public data which are associated with delay and extent of the delay

Methods – Data set

- Data was retrieved using the clinicaltrials.gov API
  - Date of extraction: 15. March 2023
  - 101 key words to search for 3 breast cancer types
- Data file consists of 2726 clinical trials and 49 variables
- Variables used in this dataset are variables which could have an effect on clinical trial duration
  - Countries, condition, population, drug, design, starting year

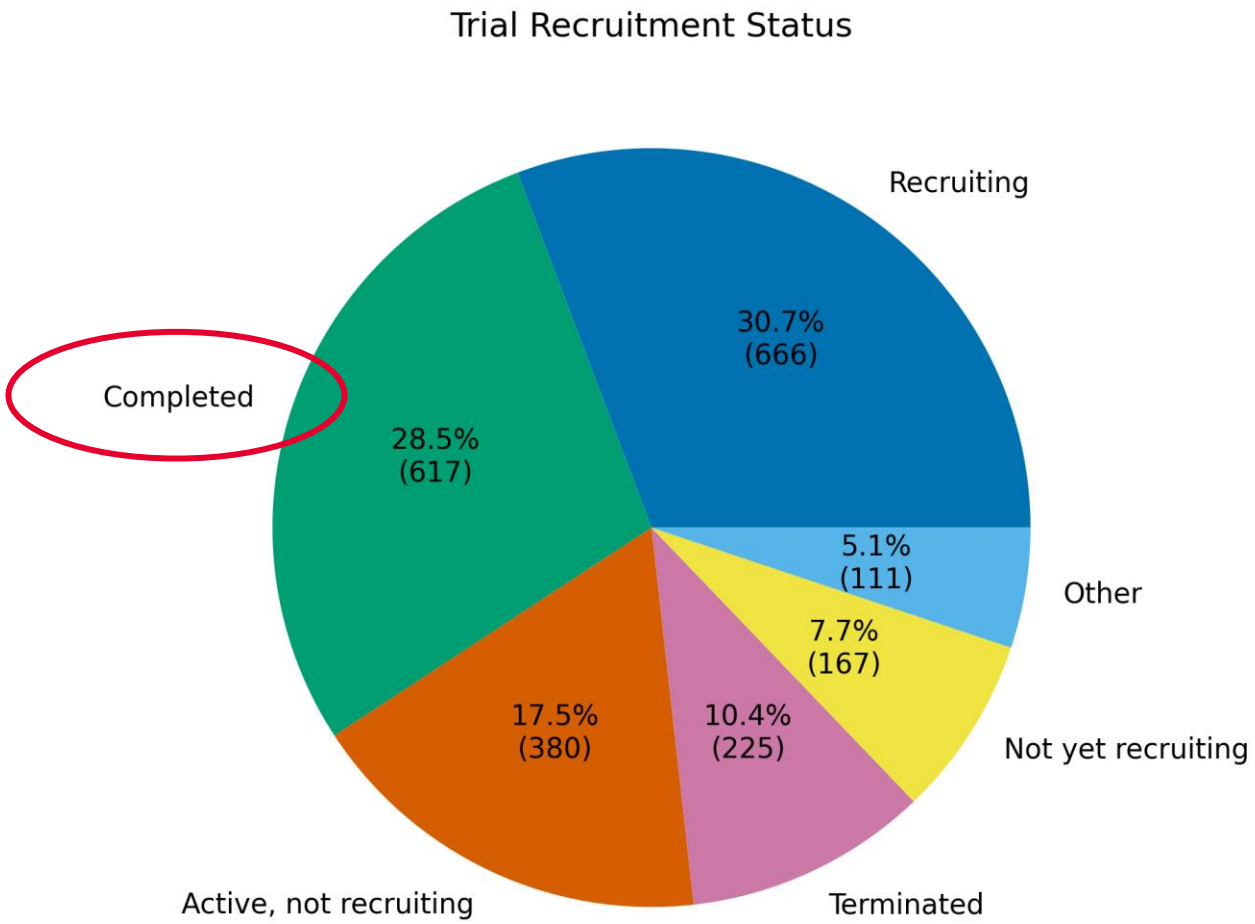
Methods - Data cleaning

- Ensure data has the right configuration
- Clean data entries
  - ['Phase 3'] → Phase 3
  - ['NIH'] → NIH
  - Rename column typo
- Filter only studies after 2005
  - Studies before 2006 do not have all the data, data fields were added in the past years
  - Low quality of data, as registration of clinical trial was not mandatory
- After filtering 2166 trials are left

Methods – Data calculations

- Duration was calculated in months:
  - Estimated duration: # of days between *Completion v1* and *StartDate* /30.44 (Average number of days per month)
  - Actual duration: # of days between *CompletionDate* and StartDate /30.44 (Average number of days per month)
- Delay was calculated as:
  - Delay in months: difference between actual and estimated duration
  - Delay in %: Delay in months as a proportion of estimated duration

Descriptive analysis - Trial



Outliers

```
# Show the 10 lowest values of DelayCalc
lowest_10 = comp_trials[['DelayCalc']].nsmallest(10, 'DelayCalc')
print(lowest_10)
```

```
      DelayCalc
1956      -174.6
1406       -74.8
1407       -74.8
790        -71.4
1355       -69.0
965        -67.0
382        -50.0
1553       -49.0
741        -48.7
1206       -48.5
```

```
# Show the 10 lowest values of DelayPerc
lowest_10 = comp_trials[['DelayPerc']].nsmallest(10, 'DelayPerc')
print(lowest_10)
```

```
      DelayPerc
1330    -9125.0
602     -1509.1
534        -88.2
1956       -85.9
347        -84.6
965        -79.8
1133       -77.6
1355       -76.7
2035       -71.7
790        -69.3
```

```
# Show the 10 largest values of DelayCalc
largest_10 = comp_trials[['DelayCalc']].nlargest(10, 'DelayCalc')
print(largest_10)
```

```
     DelayCalc
219      155.2
125      147.2
126      147.2
158      142.5
380      126.0
165      118.2
166      118.2
41       109.1
149       96.4
586       94.6
```

```
# Show the 10 largest values of DelayCalc
largest_10 = comp_trials[['DelayPerc']].nlargest(10, 'DelayPerc')
print(largest_10)
```

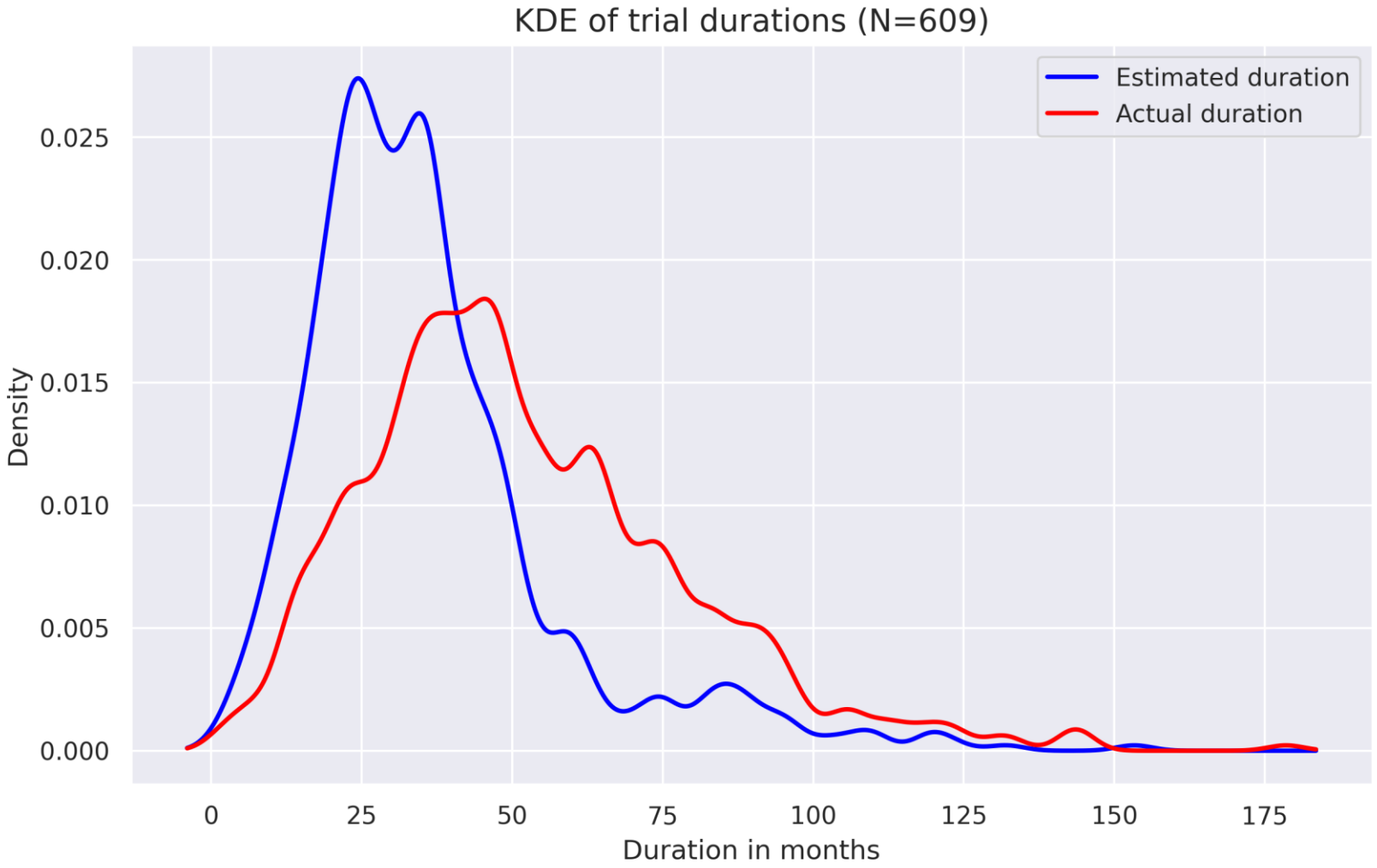
```
      DelayPerc
1289     1500.0
125      1177.6
126      1177.6
219      1175.8
856      1018.8
380       741.2
1073      685.0
193       683.3
208       625.0
628       487.5
```

Descriptive analysis – Duration

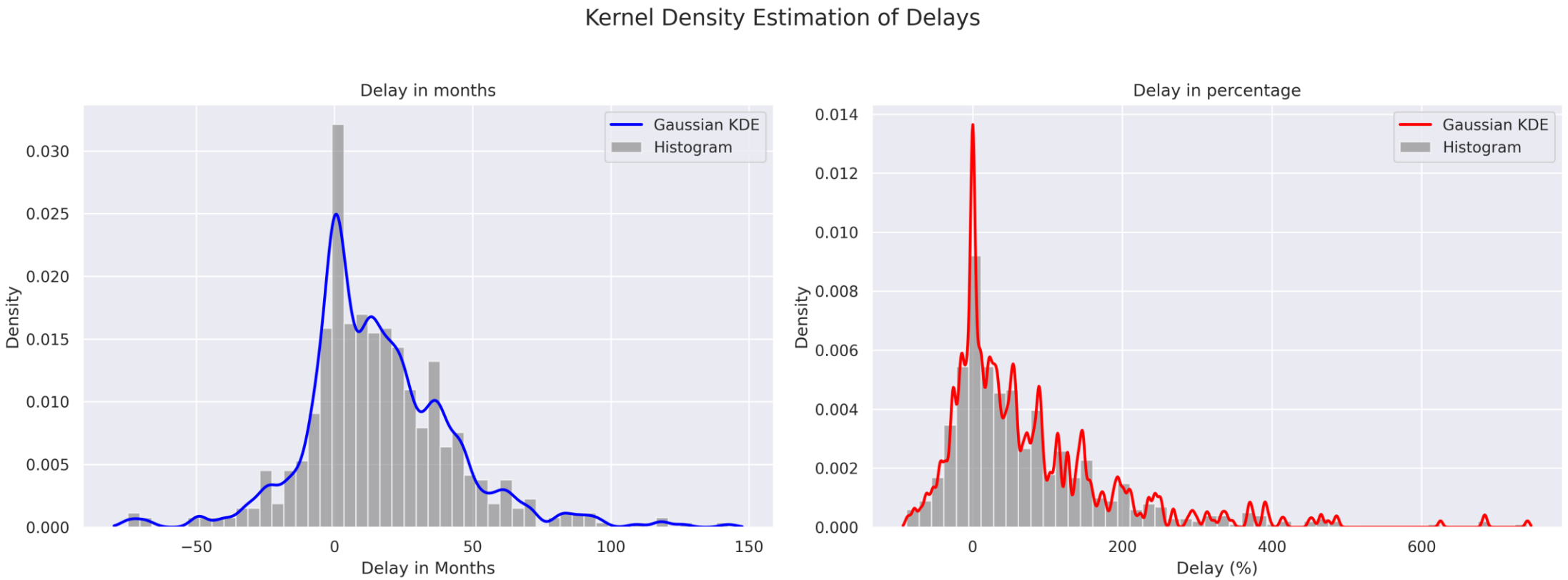
Without outliers

N=609	Estimated Duration	Actual Duration	Delay in months	Delay in percentage
mean	36.047455	51.765517	15.718062	74.083087
std	21.310501	26.352825	27.082965	115.217335
min	1.000000	1.300000	-74.800000	-88.200000
25%	23.000000	34.000000	0.000000	0.000000
50%	32.100000	47.000000	12.200000	39.300000
75%	43.000000	66.300000	30.000000	114.300000
max	153.300000	178.500000	142.500000	741.200000

Kernel Density estimation for duration



Kernel Density Estimation for delay



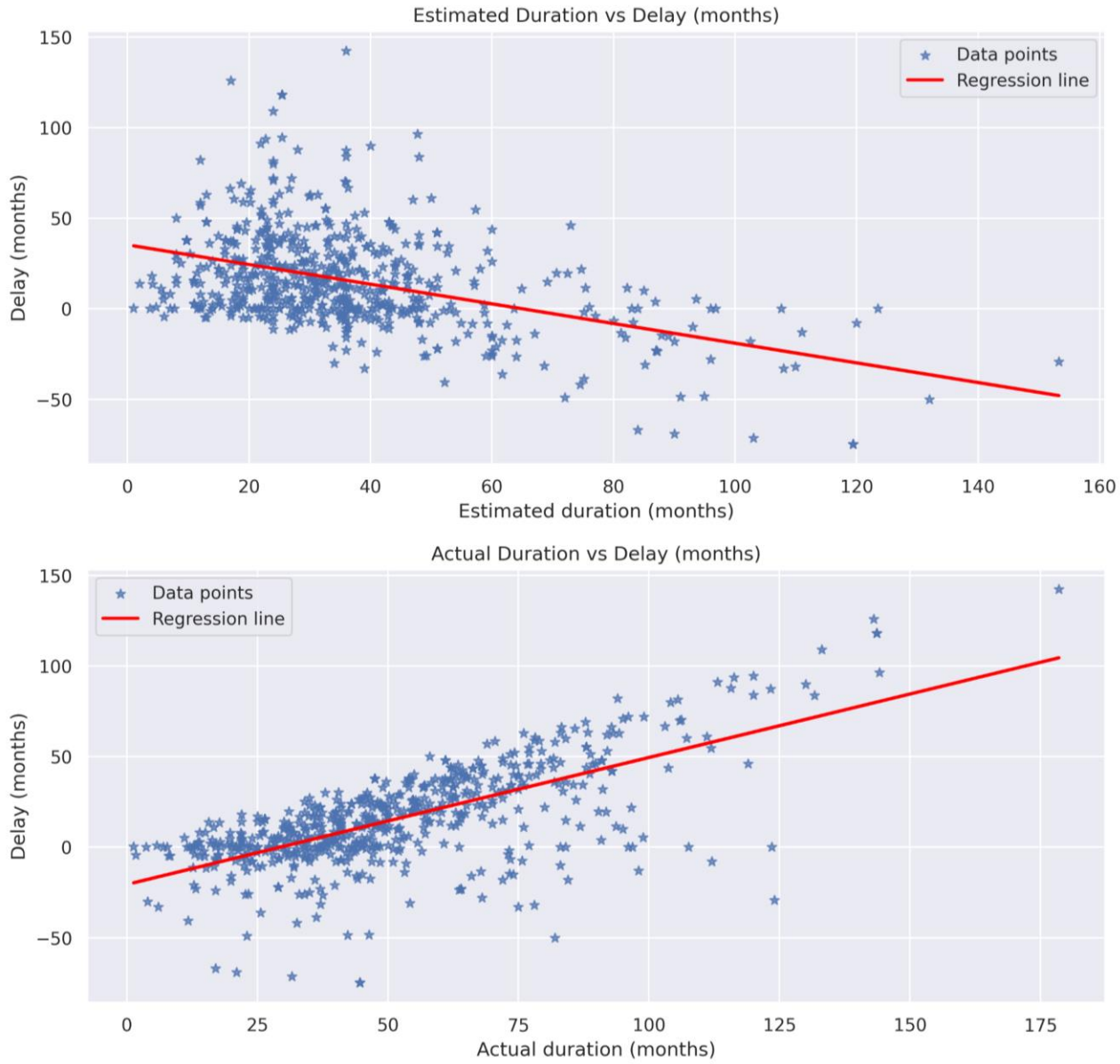
Machine Learning

- Linear regression:
  - Idea that longer trials have more time to accumulate delay
  - Longer trials means more delay

Linear regression

Left:
Estimated duration → Delay (months):
slope = -0.5430
intercept = 35.2902
MSE = 598.6223
RMSE = 24.4668
R^2 = 0.1825

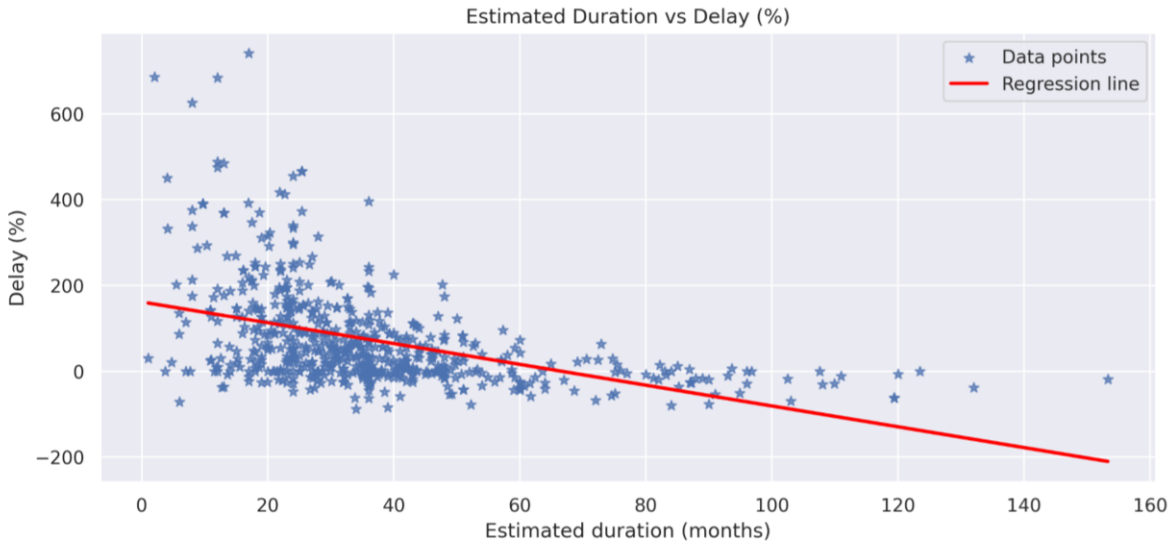
Actual duration → Delay (months):
slope = 0.7011
intercept = -20.5760
MSE = 391.4586
RMSE = 19.7853
R^2 = 0.4654

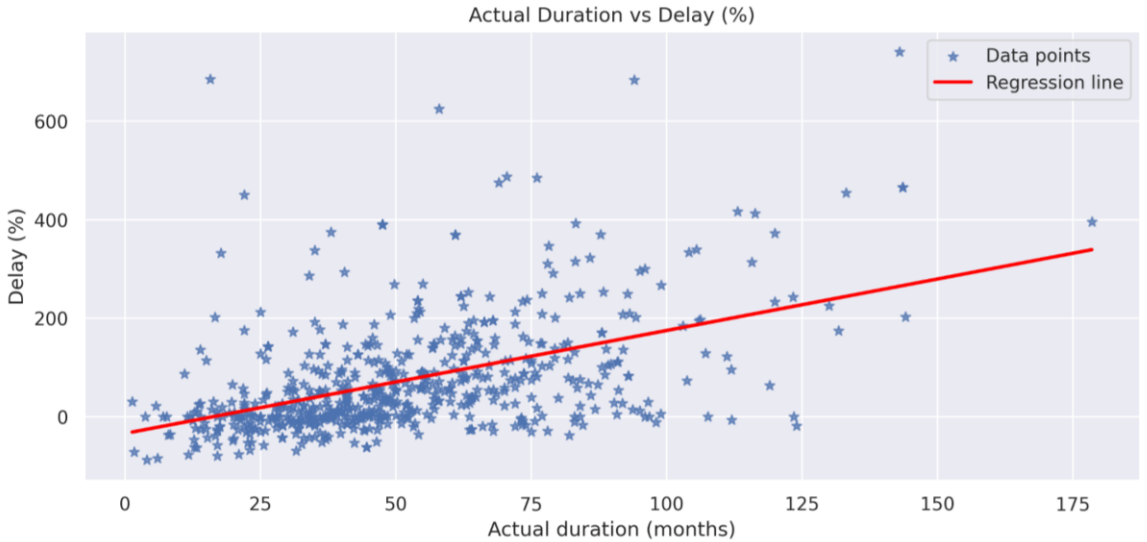


Linear regression

Estimated duration → Delay (%):
- slope = -2.4234
- intercept = 161.4409
- MSE = 10590.4994
- RMSE = 102.9102
- R^2 = 0.2009

Actual duration → Delay (%):
- slope = 2.0910
- intercept = -34.1604
- MSE = 10221.6982
- RMSE = 101.1024
- R^2 = 0.2287





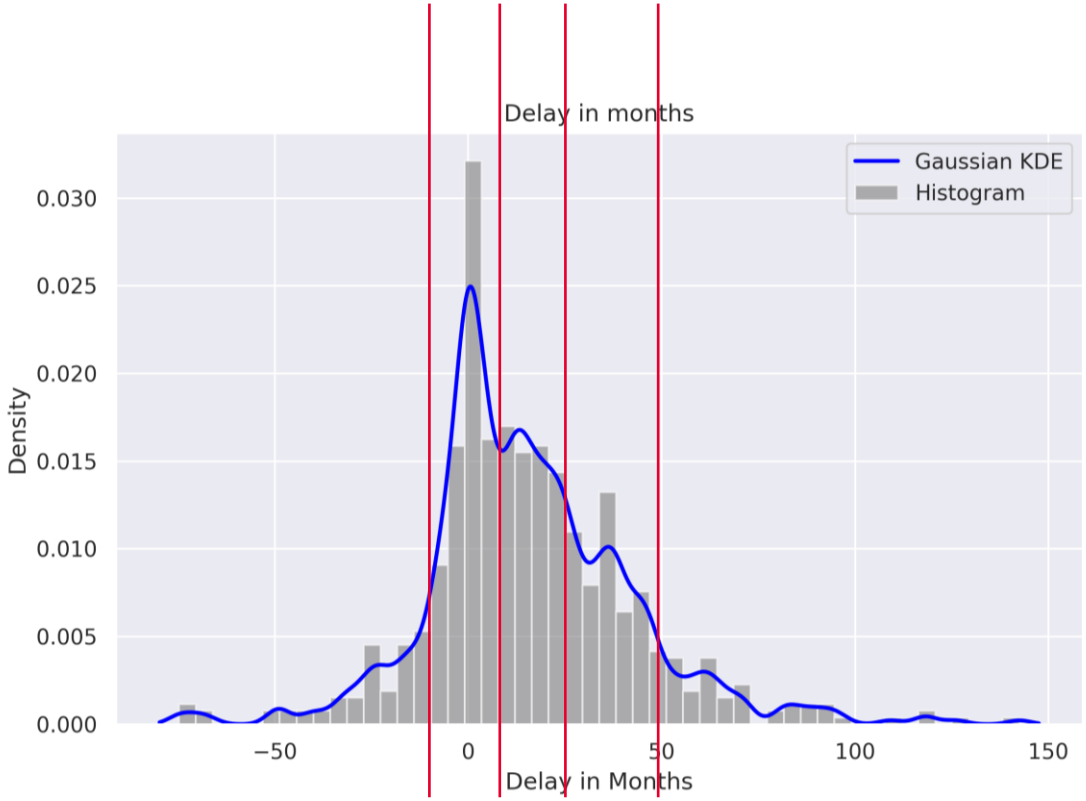
Linear regression - Conclusion

- Delay (months) has a better linear relation to project duration than Delay (%)
  - Month-delay models have lower RMSE and higher R^2.
  - Percentage delay introduces extra noise, especially when original estimates differ in scale.
- Actual duration explains delay better than estimated duration
  - R^2: 0.47 (actual) vs. 0.18–0.20 (estimated)
  - This confirms: delays are driven by what actually happens during the project, not by how long the project is supposed to take.
- Linear regression is not capturing the real structure
  - Low R^2 values suggest:
    - Delay is multi-factor.
    - Relationships are likely non-linear.
    - You need more variables (features) to explain the behavior.
    - This aligns with why your Random Forest performs better.

Machine Learning

- Random forest
  - Set delay groups
  - Numerical and categorical features

Delay groups



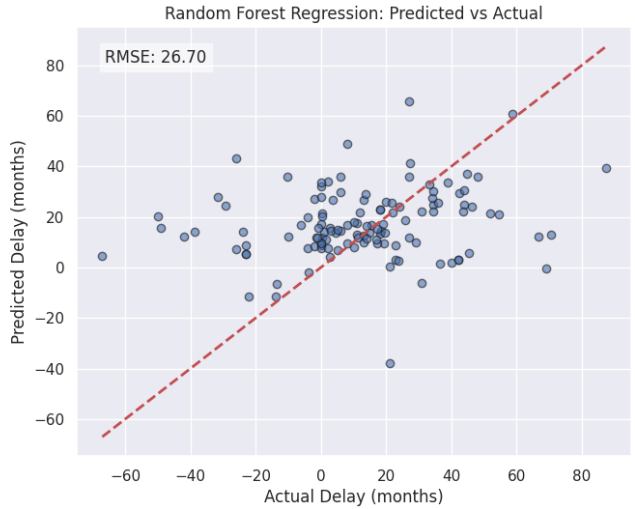
- Early: =< -3 months
- On time: -3 to 3 months
- Small delay: 3-12 months
- Large delay : 12-48 months
- Very large delay: +48 months

Features

Numerical:
- Estimated Target size
- Minimum age
- Maximum age
- Number of Inclusion criteria
- Number of Exclusion criteria
- Number of conditions
- Number of interventions
- Number of primary outcomes
- Number of secondary outcomes
- Number of arms
- Number of Enrolled participants

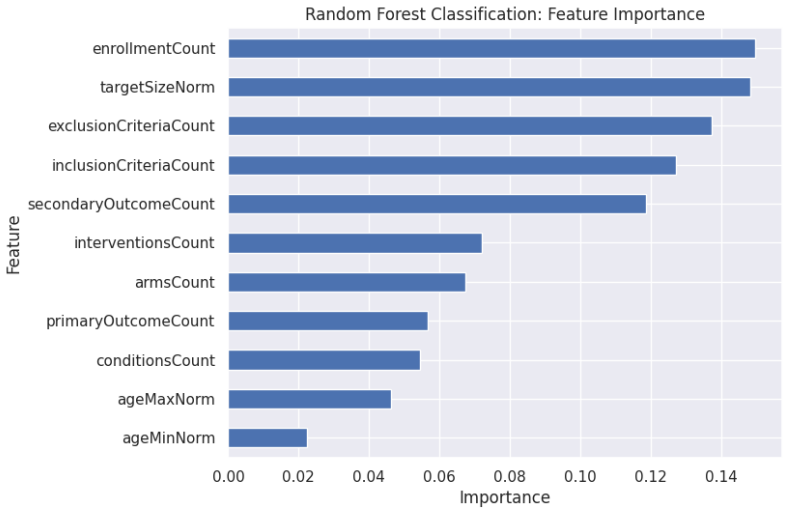
Categorical:
- Gender
- Phase
- Organisation class
- Design
- Breast cancer group

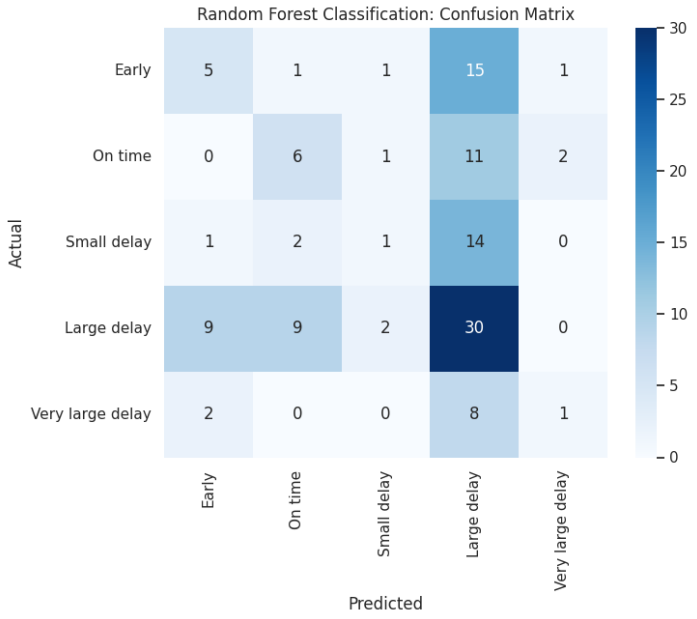
Random forest Delay in months



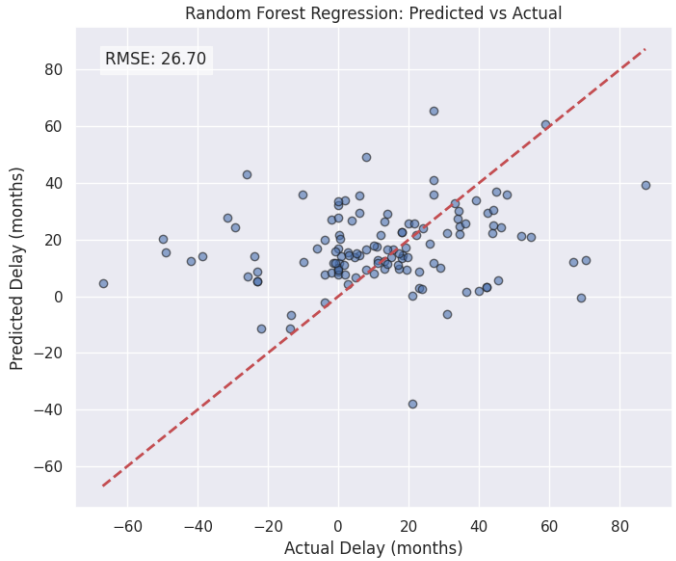
Classification Report:

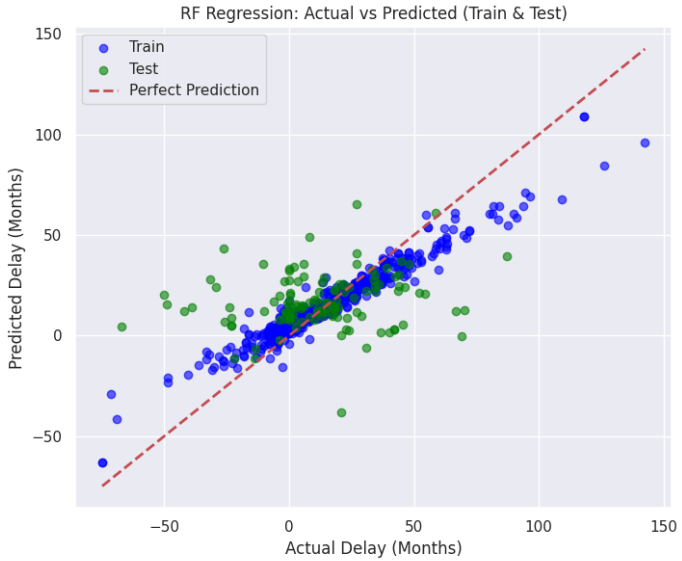
	precision	recall	f1-score	support
Early	0.29	0.22	0.25	23
On time	0.38	0.60	0.47	50
Small delay	0.33	0.30	0.32	20
Large delay	0.20	0.06	0.09	18
Very large delay	0.25	0.09	0.13	11
accuracy			0.35	122
macro avg	0.29	0.25	0.25	122
weighted avg	0.32	0.35	0.32	122





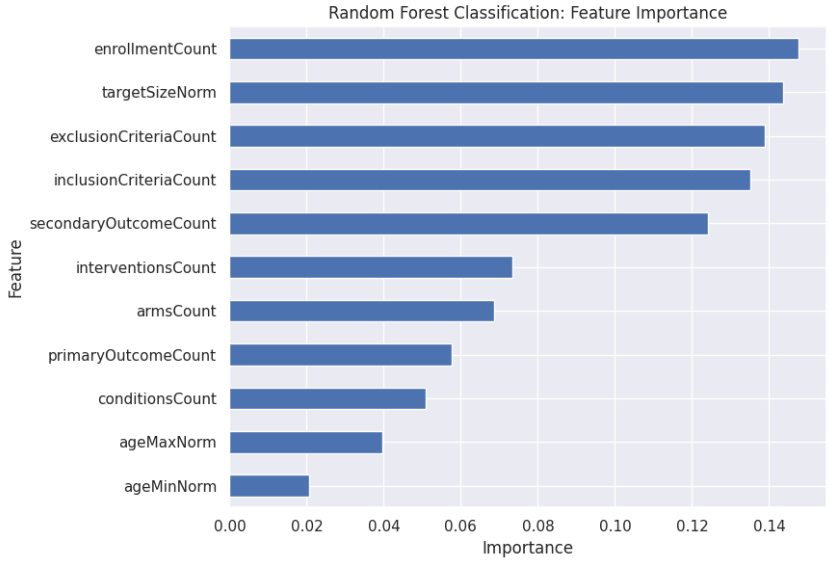
Random forest: Delay in months

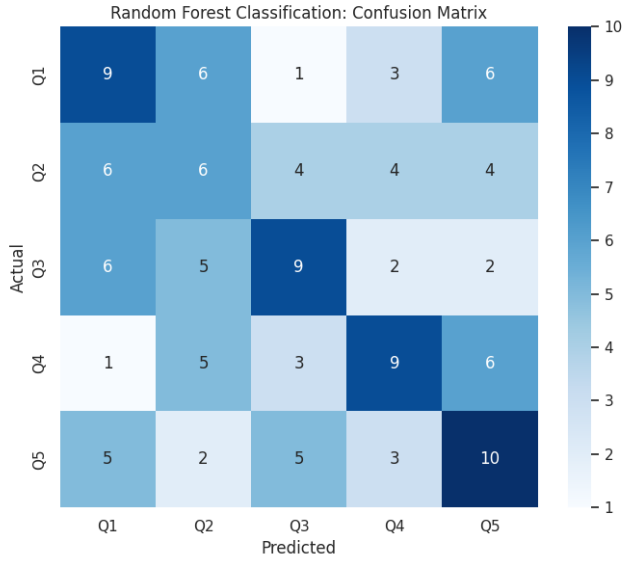




Classification Report:

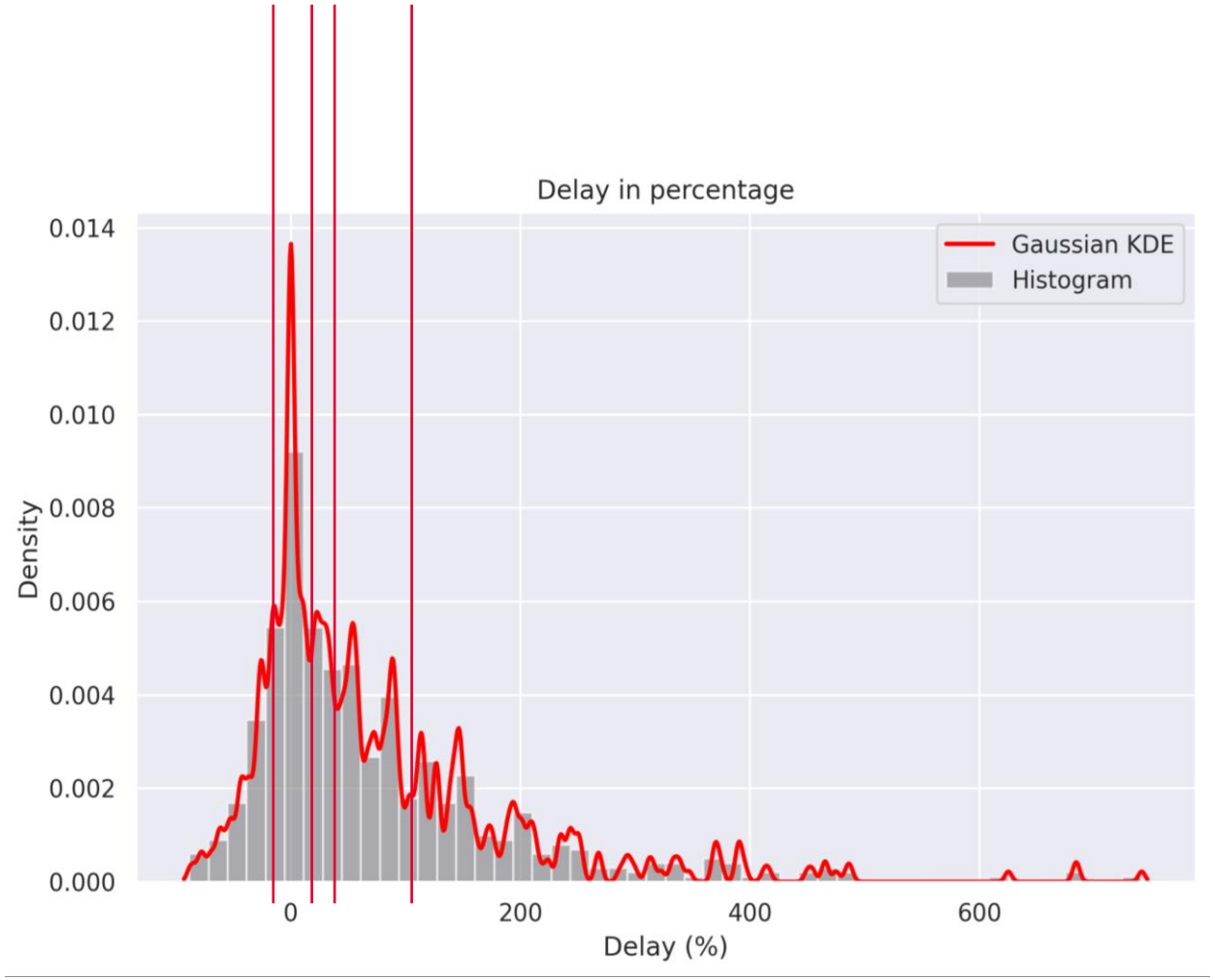
	precision	recall	f1-score	support
Q1	0.33	0.36	0.35	25
Q2	0.25	0.25	0.25	24
Q3	0.41	0.38	0.39	24
Q4	0.43	0.38	0.40	24
Q5	0.36	0.40	0.38	25
accuracy			0.35	122
macro avg	0.36	0.35	0.35	122
weighted avg	0.36	0.35	0.35	122





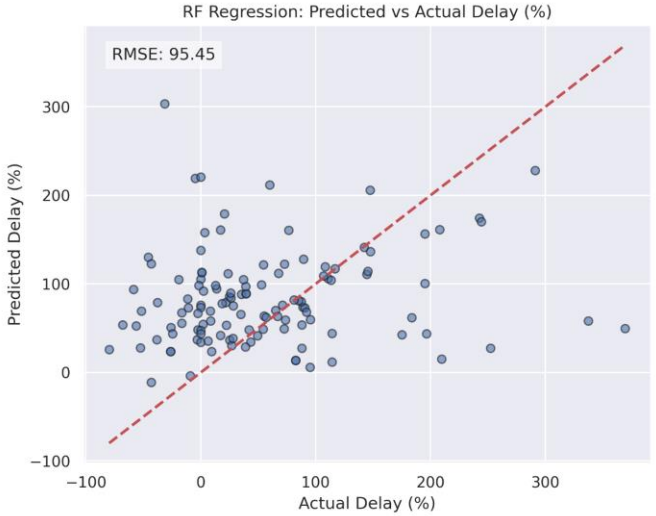
Group 1: -74.80 to -2.00 months
Group 2: -2.00 to 5.92 months
Group 3: 5.92 to 18.56 months
Group 4: 18.56 to 35.82 months
Group 5: 35.82 to 142.50 months

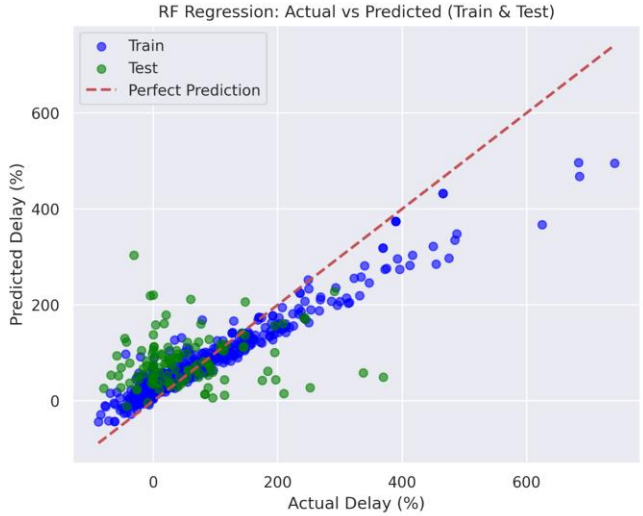
Kernel Density Estimation for delay



- Early: =< -10%
- On time: -10% - 10%
- Small delay:10% - 50%
- Large delay : 50% - 100%
- Very large delay: +100%

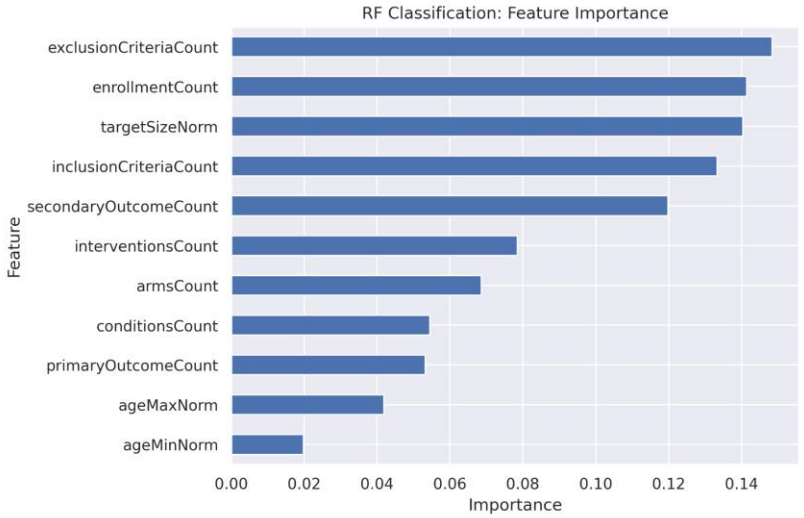
Random forest: Delay as %

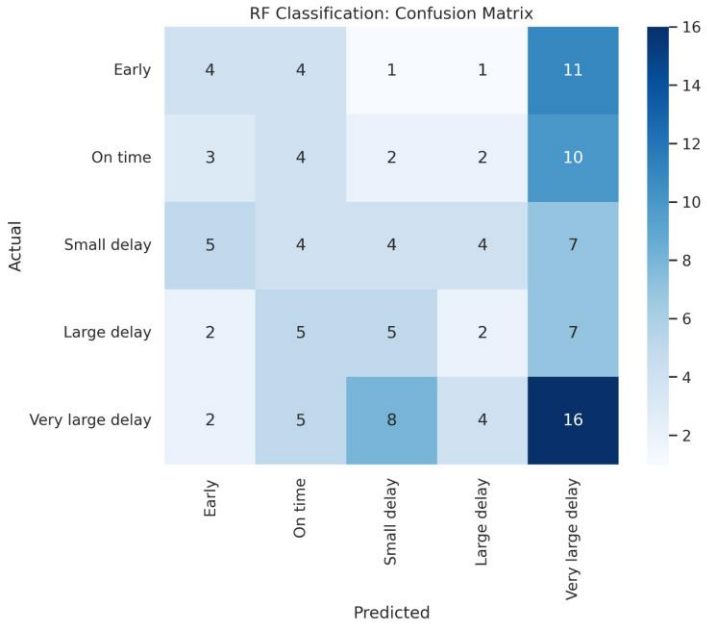




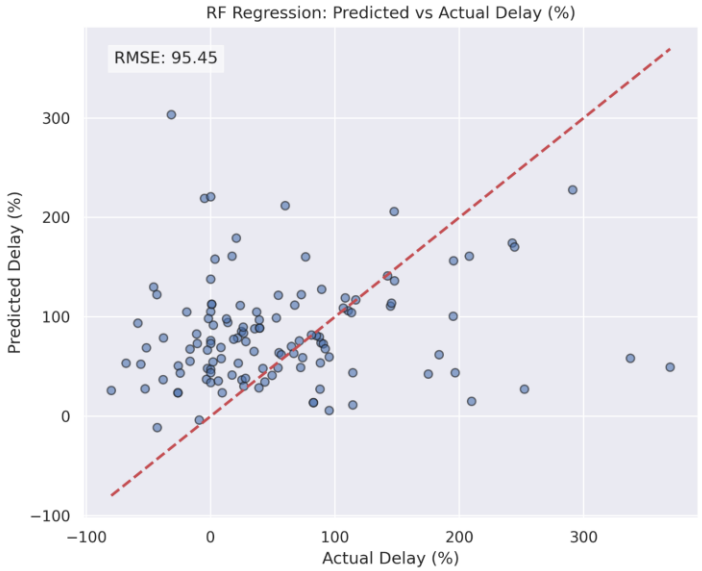
Classification Report (Delay %):

	precision	recall	f1-score	support
Early	0.25	0.19	0.22	21
On time	0.15	0.10	0.12	21
Small delay	0.18	0.19	0.19	21
Large delay	0.20	0.17	0.18	24
Very large delay	0.31	0.46	0.37	35
accuracy			0.25	122
macro avg	0.22	0.22	0.21	122
weighted avg	0.23	0.25	0.23	122

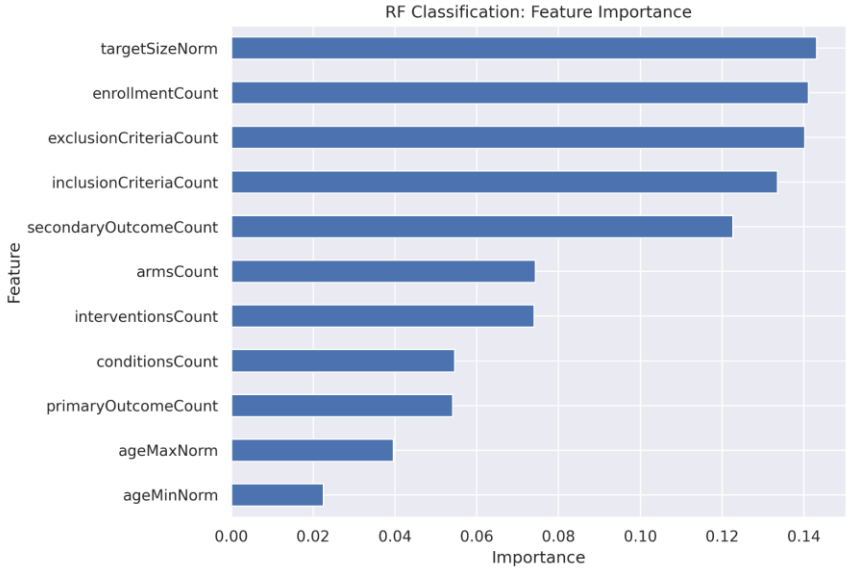


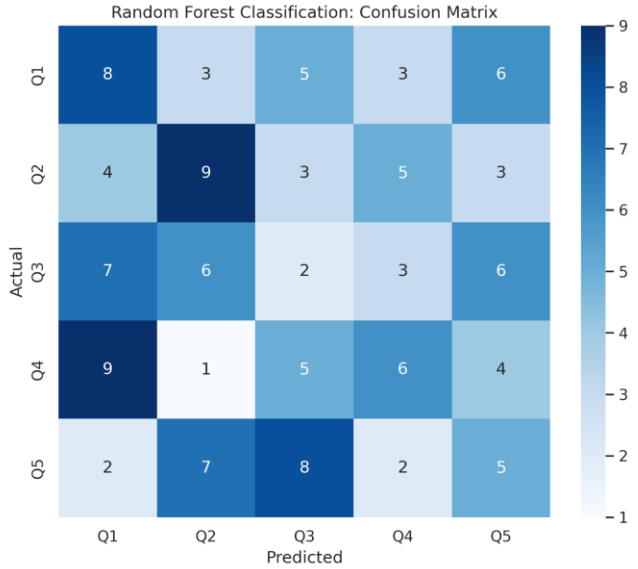


Random forest: Delay as %



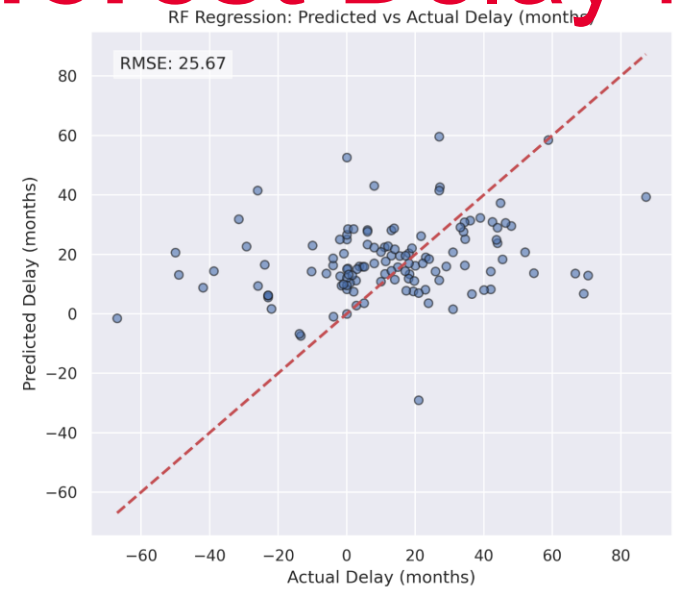
	precision	recall	f1-score	support
Q1	0.27	0.32	0.29	25
Q2	0.35	0.38	0.36	24
Q3	0.09	0.08	0.09	24
Q4	0.32	0.24	0.27	25
Q5	0.21	0.21	0.21	24
accuracy			0.25	122
macro avg	0.24	0.25	0.24	122
weighted avg	0.25	0.25	0.24	122

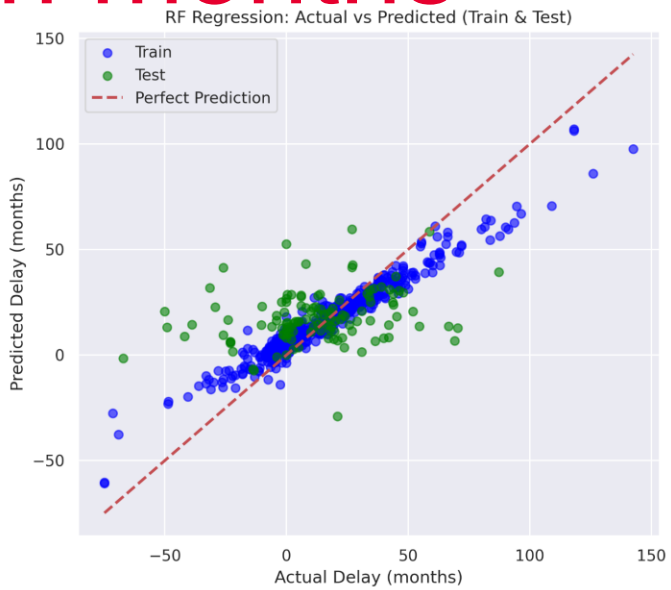


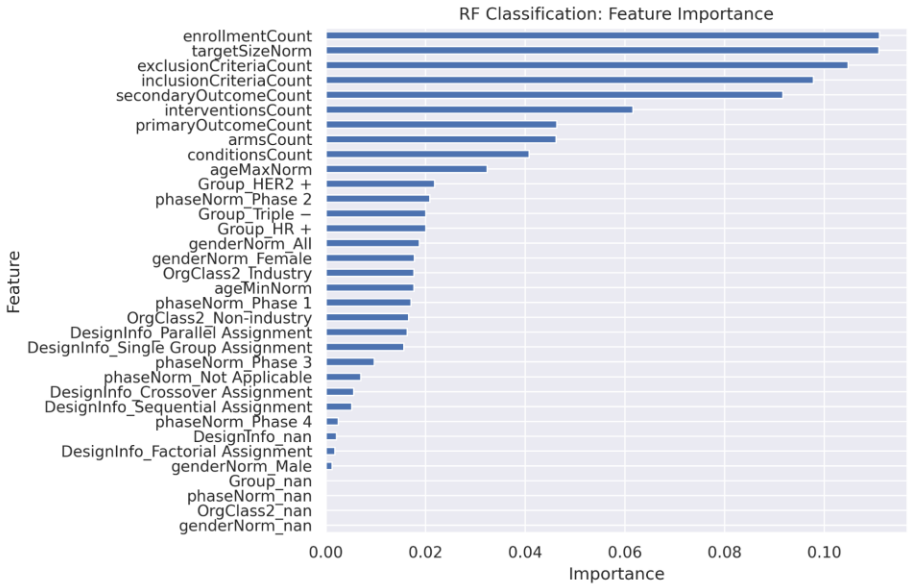


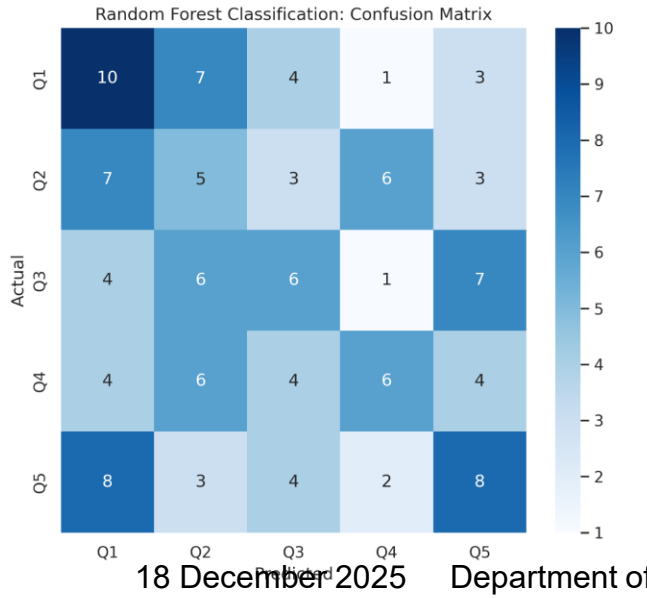
Bin edges for DelayPattern groups:
Group 1: -88.20 to -5.24 %
Group 2: -5.24 to 20.52 %
Group 3: 20.52 to 62.90 %
Group 4: 62.90 to 141.60 %
Group 5: 141.60 to 741.20 %

Random forest Delay in months

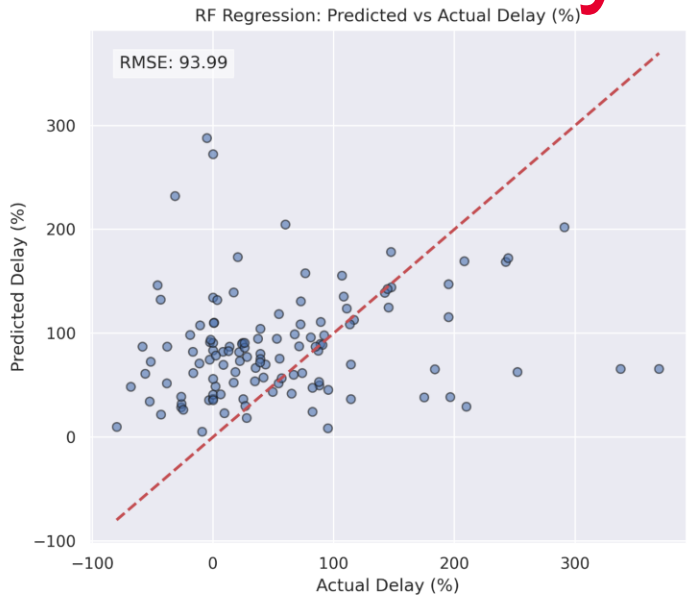


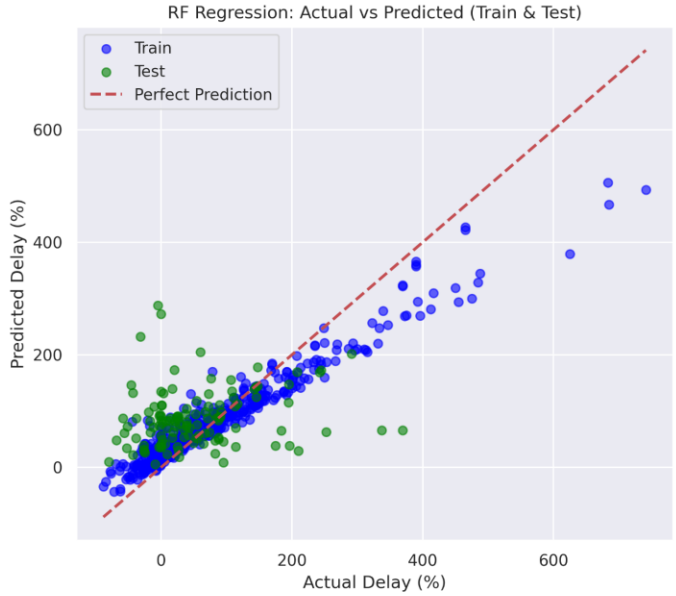


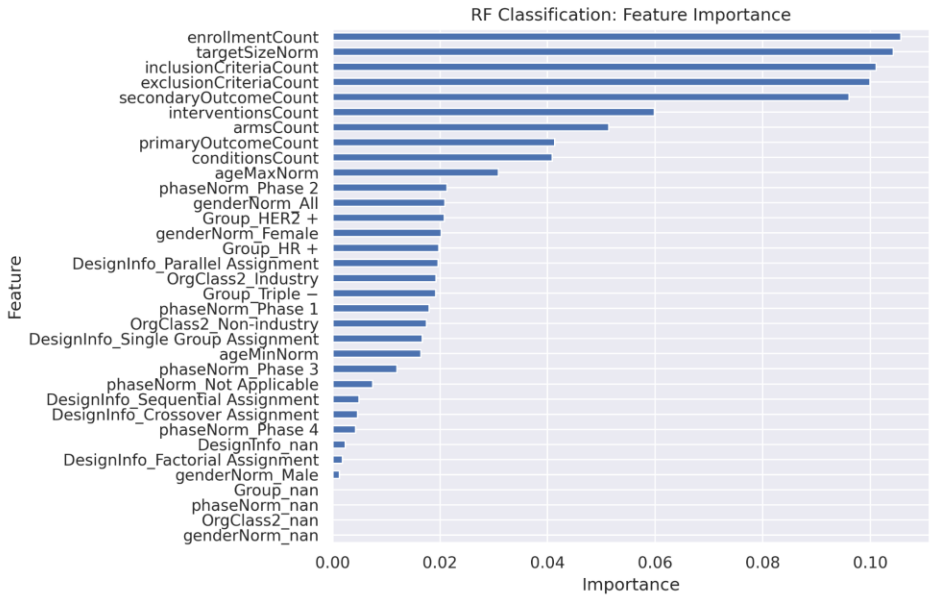


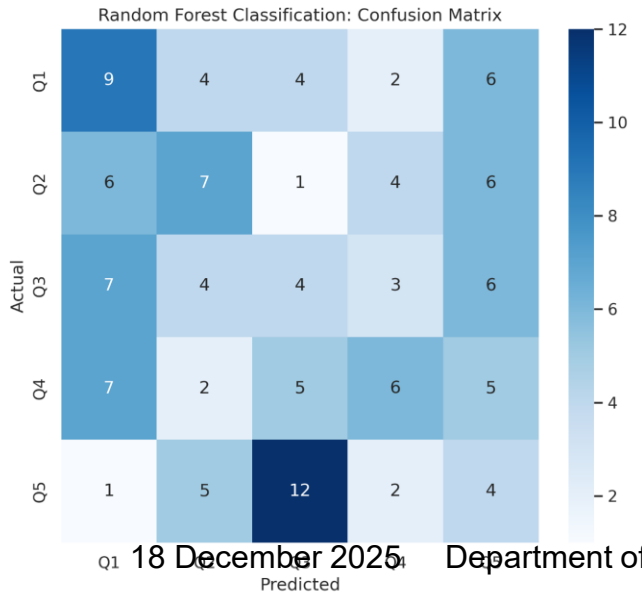


Random forest Delay in %





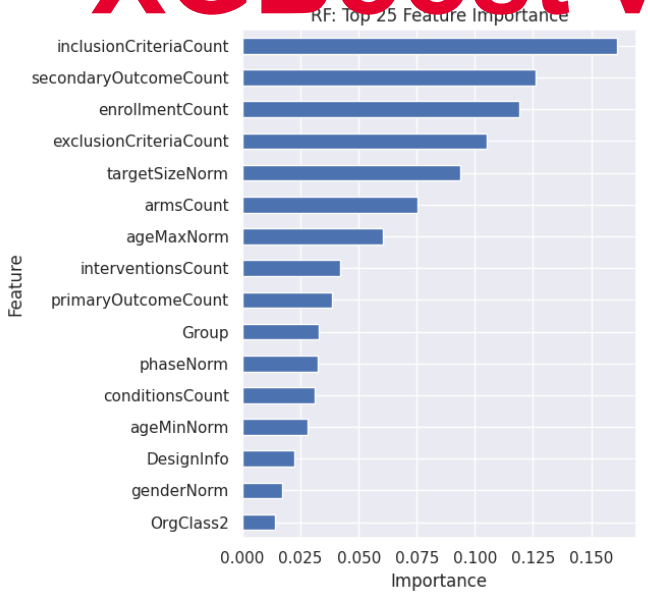


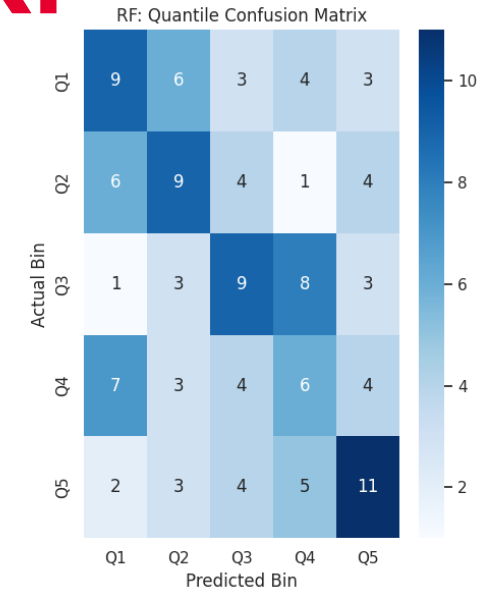


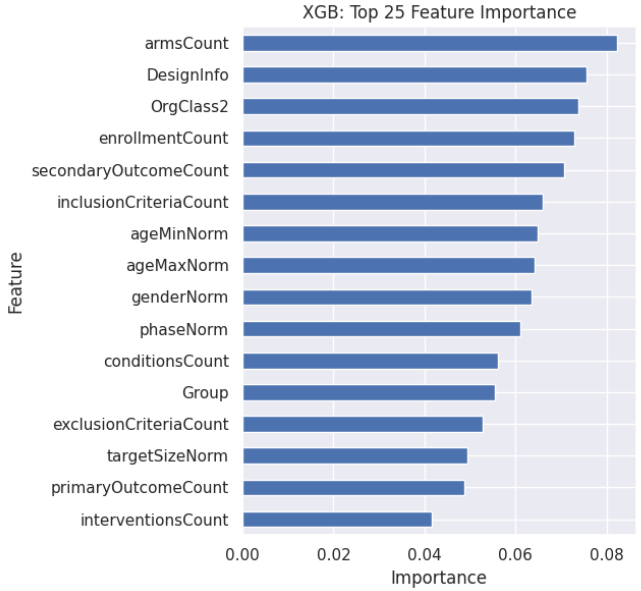
Random forest

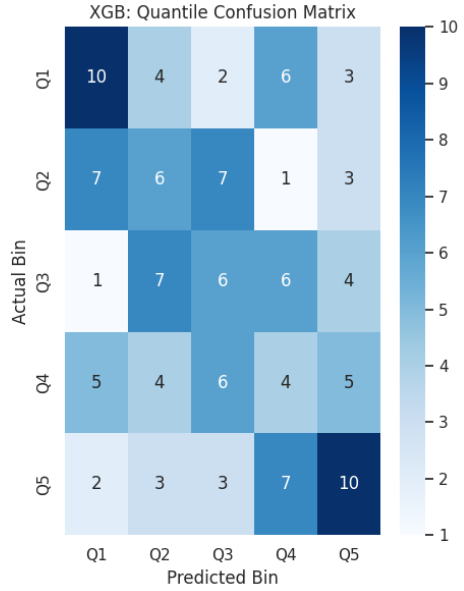
- This model sucks…
- Low performance overall
  - Accuracy 25-35% and F1-scores are low
- Favors majority classes
- Class separability is weak → Might be due to the small groups.

XGBoost vs RF









Random Forest Quantile Classification Report:

	precision	recall	f1-score	support
Q1	0.36	0.36	0.36	25
Q2	0.38	0.38	0.38	24
Q3	0.38	0.38	0.38	24
Q4	0.25	0.25	0.25	24
Q5	0.44	0.44	0.44	25
accuracy			0.36	122
macro avg	0.36	0.36	0.36	122
weighted avg	0.36	0.36	0.36	122

XGBoost Quantile Classification Report:

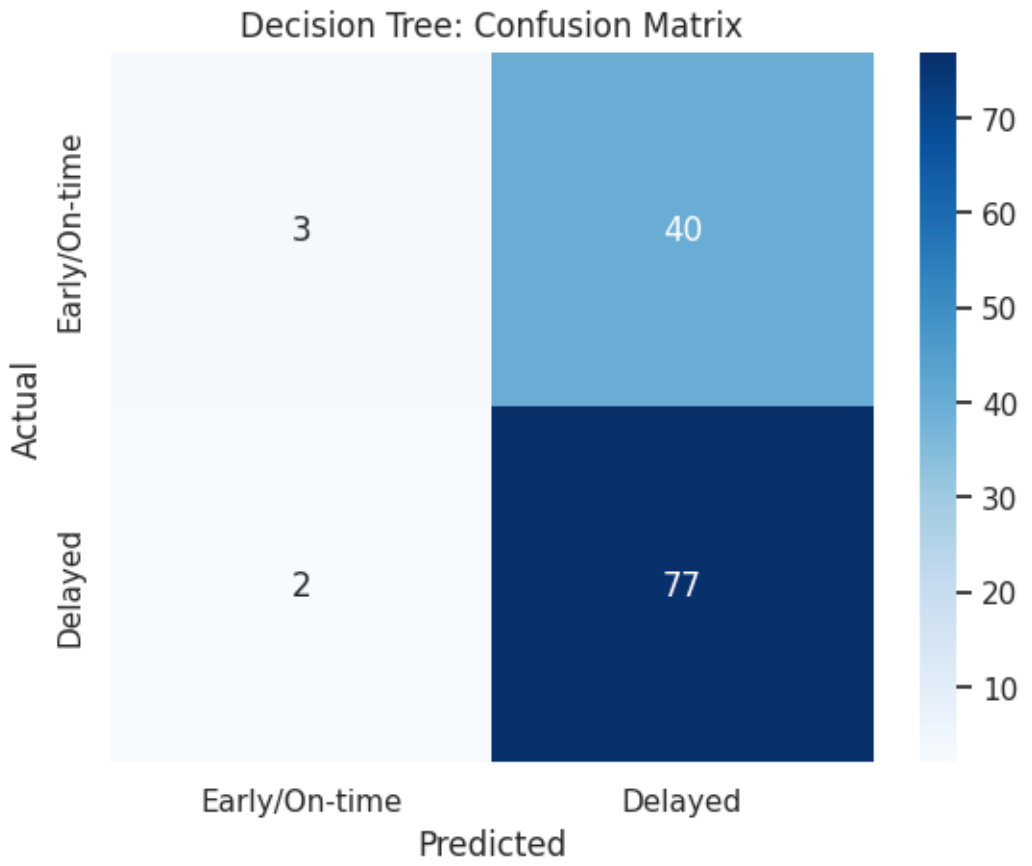
	precision	recall	f1-score	support
Q1	0.40	0.40	0.40	25
Q2	0.25	0.25	0.25	24
Q3	0.25	0.25	0.25	24
Q4	0.17	0.17	0.17	24
Q5	0.40	0.40	0.40	25
accuracy			0.30	122
macro avg	0.29	0.29	0.29	122
weighted avg	0.30	0.30	0.30	122

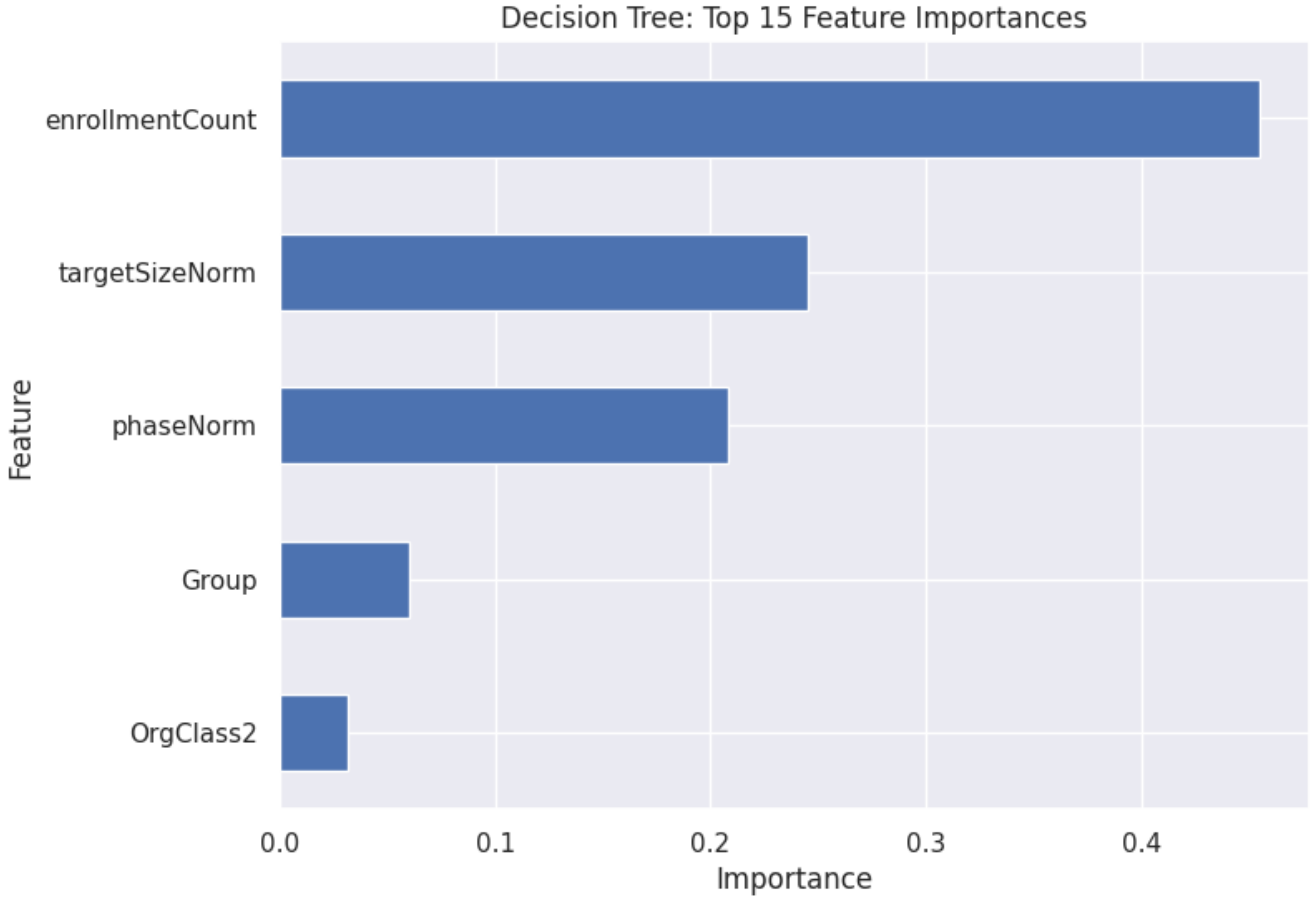
RF and XGBoost

- **Best-performing model:** Random Forest
- More balanced F1 across classes
- Handles middle quantiles better than XGBoost
- **Weakest model:** XGBoost for quantile / % classification
- Strongly favors extreme classes, poor on middle delays
- **Easiest classes to predict:**
- Extreme quantiles (Q1, Q5)
- Delayed in binary classification
- **Hardest classes to predict:**
- Middle quantiles (Q2–Q4)
- Early / On-time in binary classification
- Q3 and Q4 in % classification
- **Key factors limiting performance:**
- Small class sizes → insufficient data per class
- Overlapping feature distributions → poor class separability
- Low signal for subtle differences in delay patterns

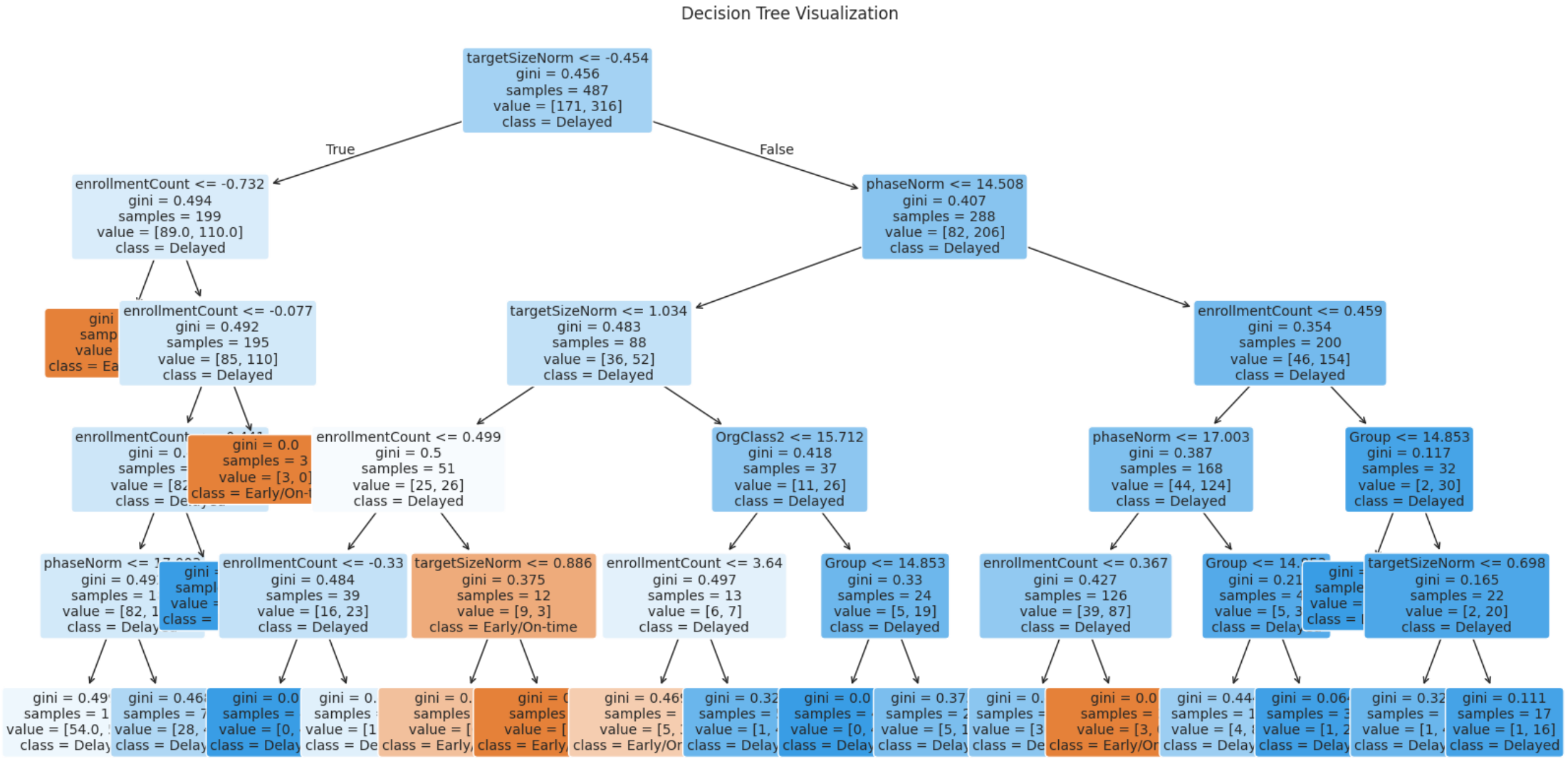
ML - Classification

Classification



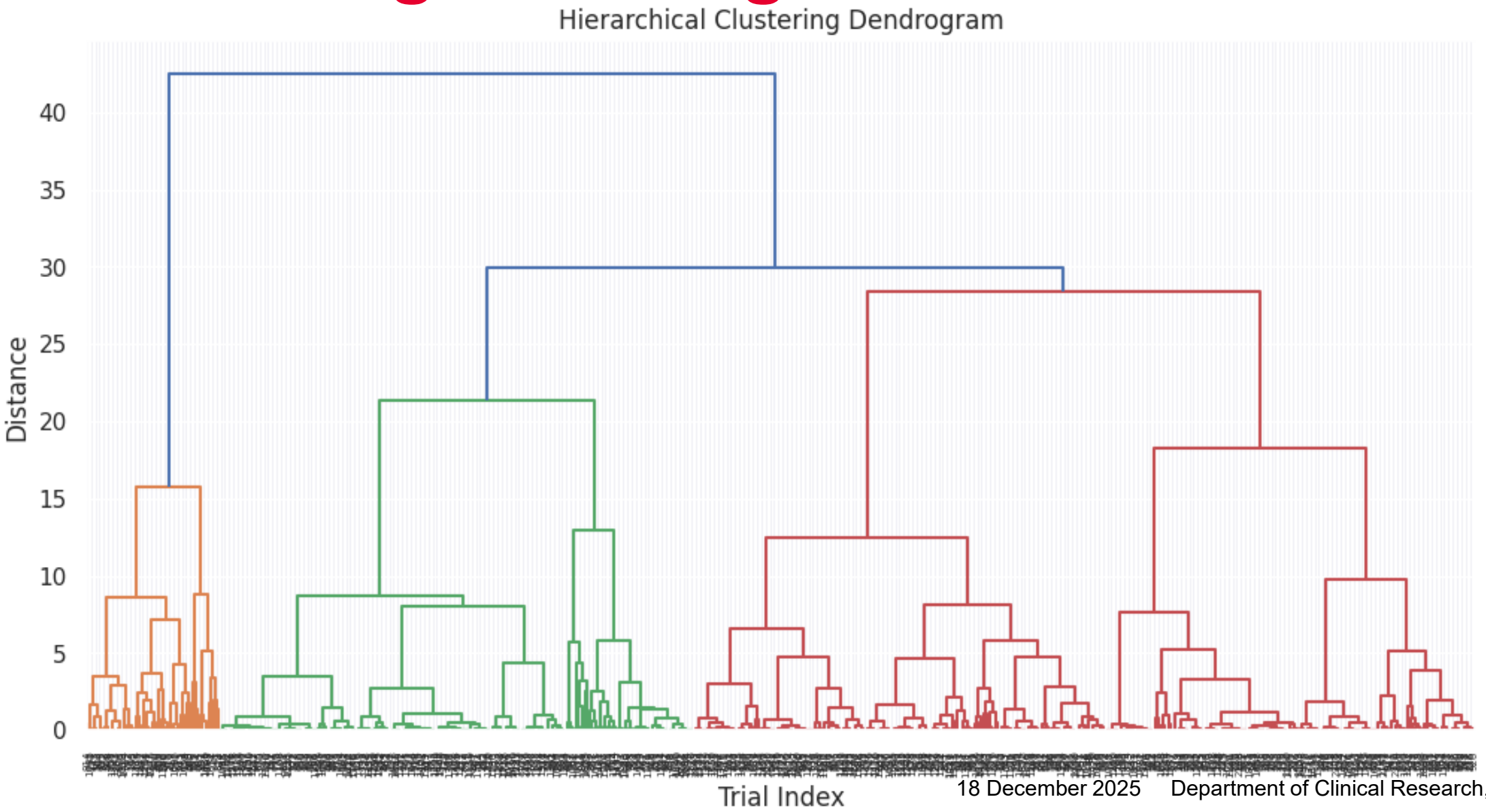


Decision tree



ML - Clustering

Clustering - Dendogram



Conclusion

- None of the models really had good performance.
- Try with bigger dataset
- There are still options to improve:
  - Feature engineering → see if other variables would give better results
  - Change modeling approach → Delays as numbers or ordered groups
  - Fix class Imbalance → Small groups properly represented
  - Tune model settings → tree depth, number of trees, learning rate