

Overview of Classification

→ **Two** separate classification analyses:

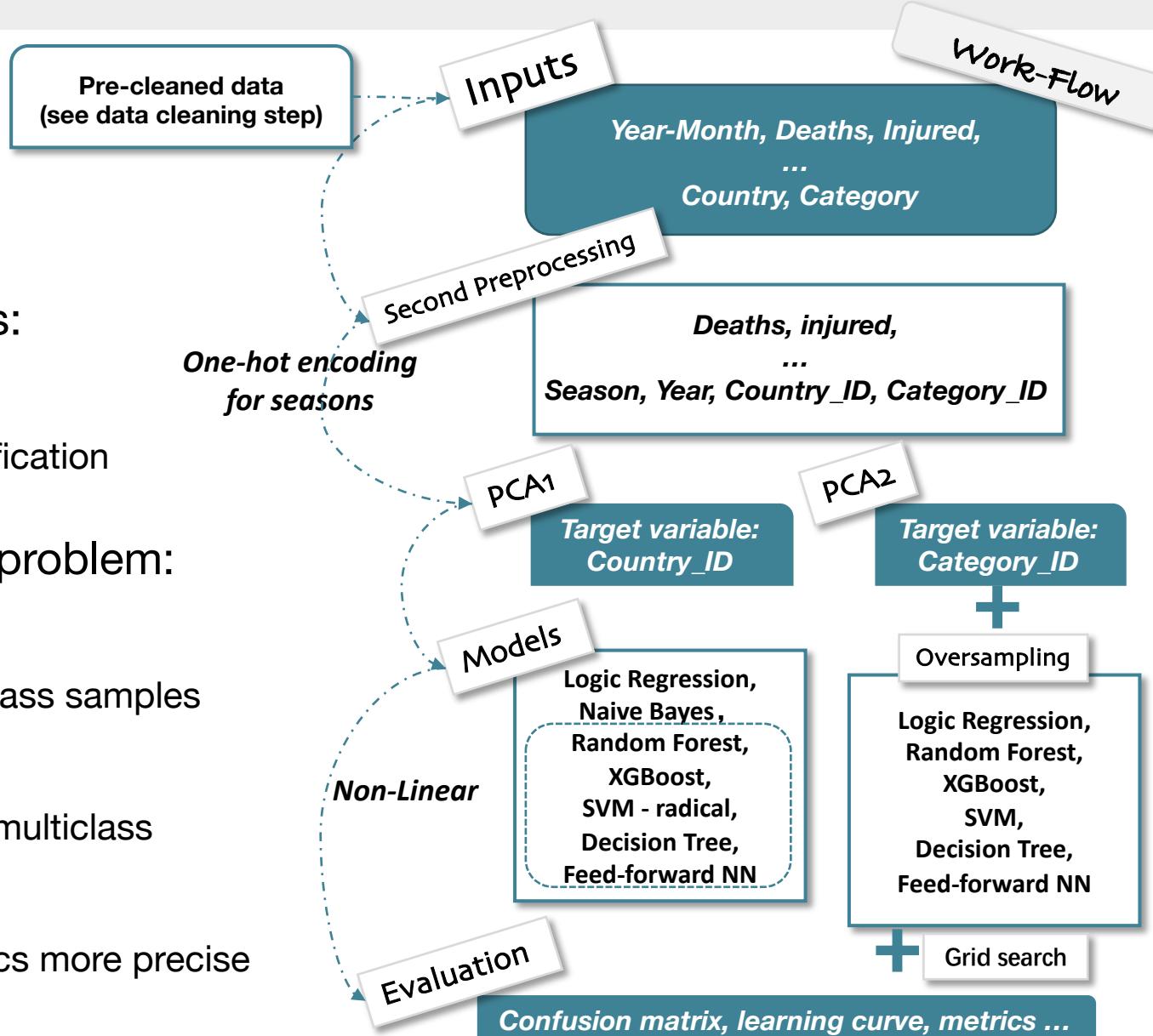
- ◆ predict country's continents - *binary* problem
- ◆ predict disaster's types - *multiclass* problem

→ **Similar** Challenges appear in two problems:

- ◆ dirty data, full of missing/wrong values
- ◆ Imbalanced data, especially for multiclass classification

→ **Various** techniques applied to tackle each problem:

- ◆ Pre-processing disaster categories
- ◆ SMOTE to generate synthetic data for minority class samples
- ◆ PCA technique applied to reduce dimensionality
- ◆ grid search to optimize the hyperparameters for multiclass problem
- ◆ 5-Fold cross-validation applied to evaluate metrics more precise



Binary classification

→ One target with two label:

- ◆ South America(1) and Africa(2)

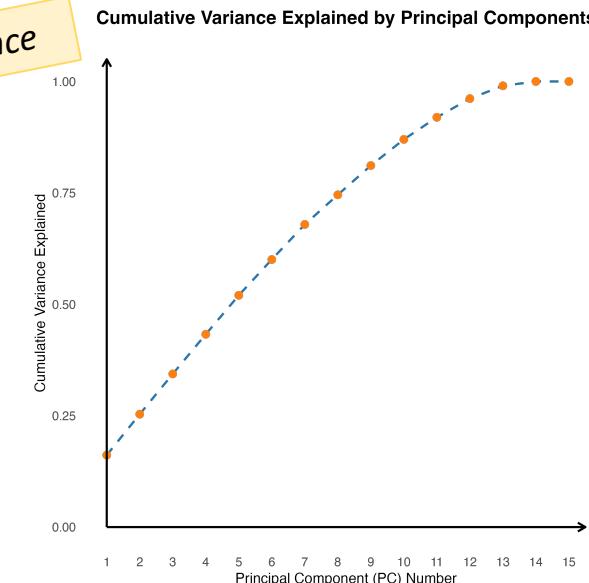
→ First 12 principal components were selected

- ◆ capture 99% of the variance
- ◆ loading plot reveals the relationship with original data [1]

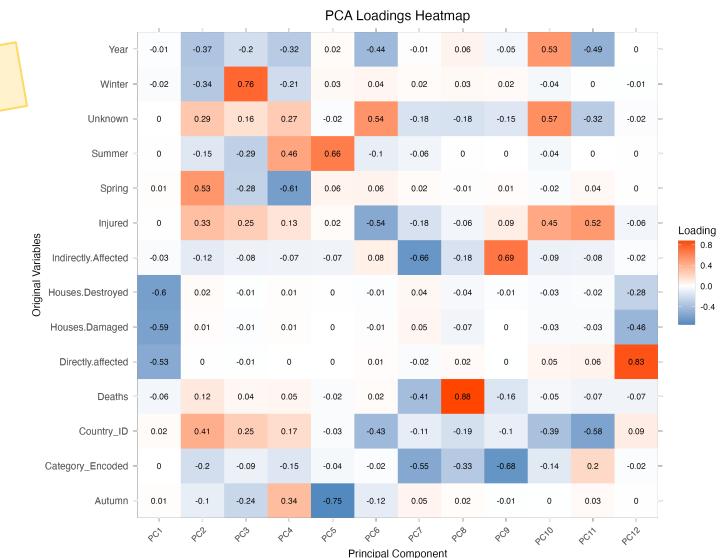
→ Various models are trained:

- ◆ Logistic Regression: A simple linear model
- ◆ Random Forest: $mtry = \sqrt{n}$
- ◆ SVM: radial kernel, cost = 1, gamma = 0.1
- ◆ XGBoost: evaluate logarithmic loss during training
- ◆ Decision Tree: CART
- ◆ FFNN: Feedforward Architecture with Single Hidden Layer
- ◆ Naïve Bayes (as a backup)

Cumulative Variance



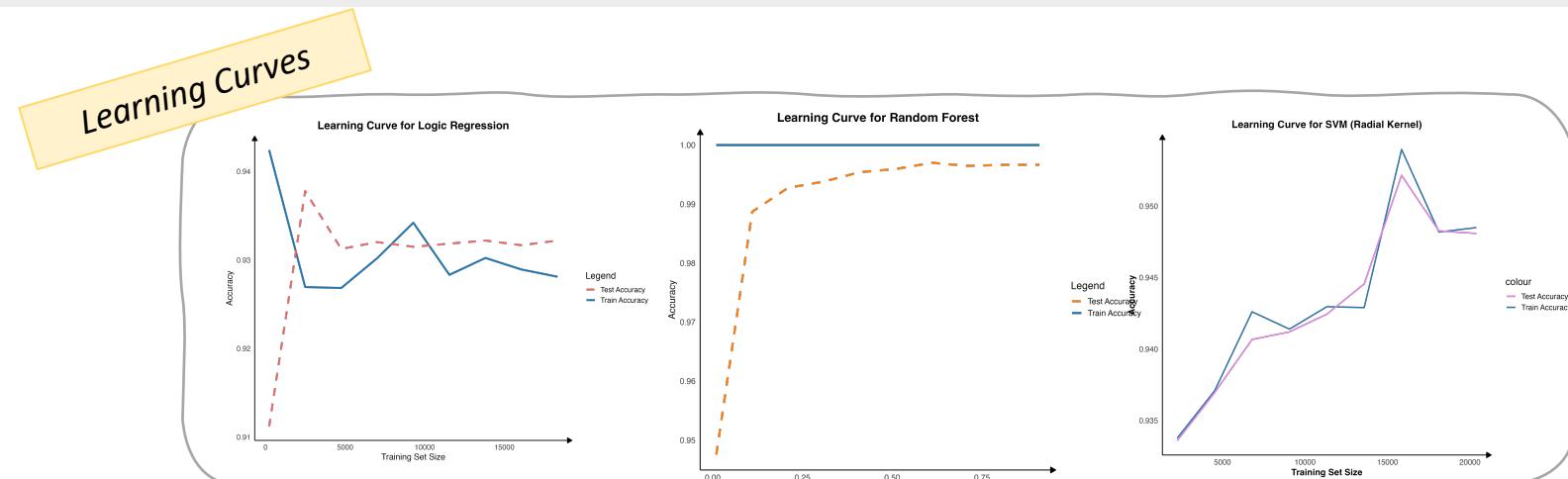
PCA Loading



Binary classification

→ Logistic Regression

- ◆ Fluctuating Test Accuracy
- ◆ Possible Underfitting at Small Sizes
- ◆ Low sensitivity and recall
- ◆ Model tend to predict as majority category



→ Random Forest

- ◆ Stable Test Accuracy
- ◆ Consistency Across Sizes
- ◆ Minimal Errors in two classes
- ◆ Consistent Performance for two classes

→ SVM - Radial Kernel

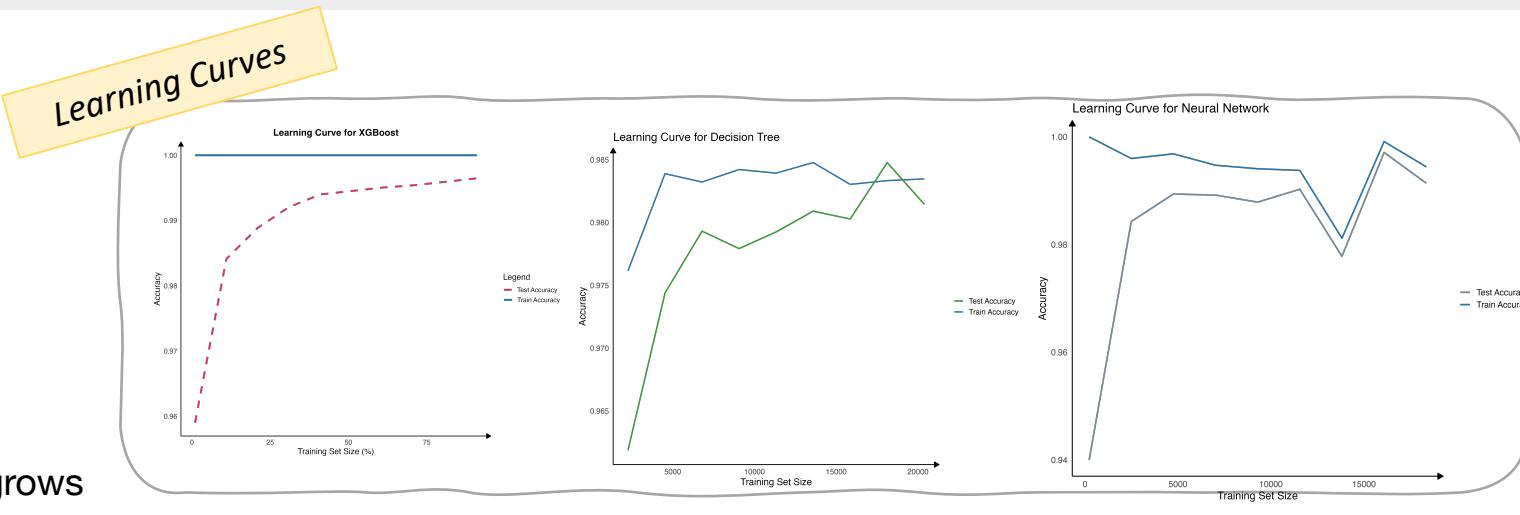
- ◆ Converging Accuracy
- ◆ High Variability at Small Sizes
- ◆ Potential Optimal Training Size Range Identified
- ◆ Higher Misclassification in Class 2
- ◆ High False Negative Events



Binary classification

→ XGboost

- ◆ Log Loss shows convergence as the number of training rounds increases
- ◆ The model generalizes better over time
- ◆ Lower false negatives and false positives



→ Decision Tree

- ◆ test and training accuracies stabilize as the size grows
- ◆ Decision trees can overfit on smaller datasets but perform more consistently with larger data
- ◆ Higher false negatives and false positives

XGboost

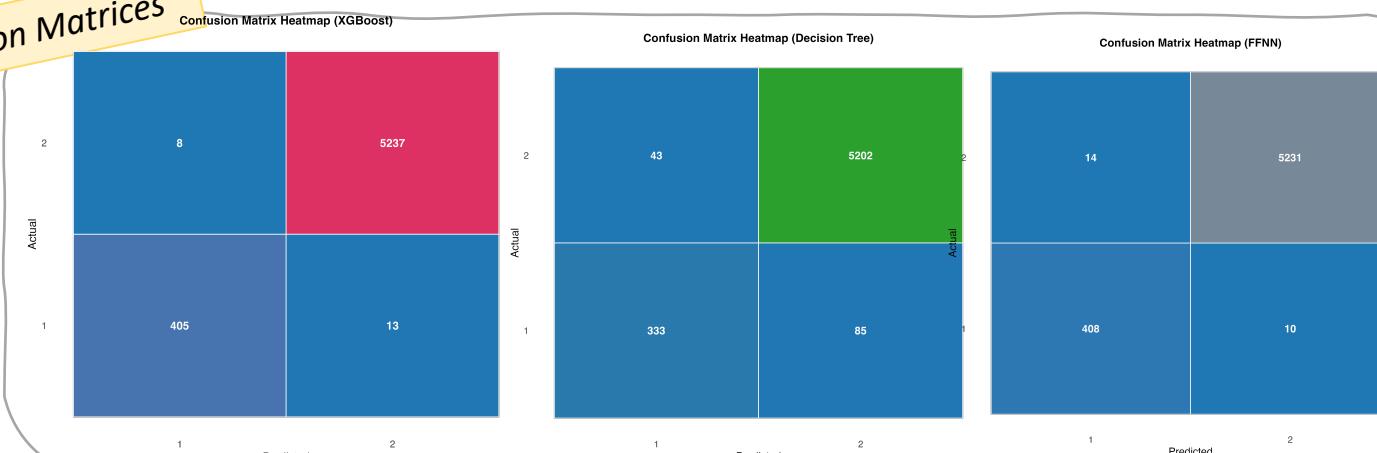
Decision Tree

Feed-forward NN

→ Feed-forward NN

- ◆ The accuracy shows fluctuations, likely due to the nonlinear training process of neural networks
- ◆ Generally good for both class1 and class2
- ◆ Acceptable false negatives and false positives

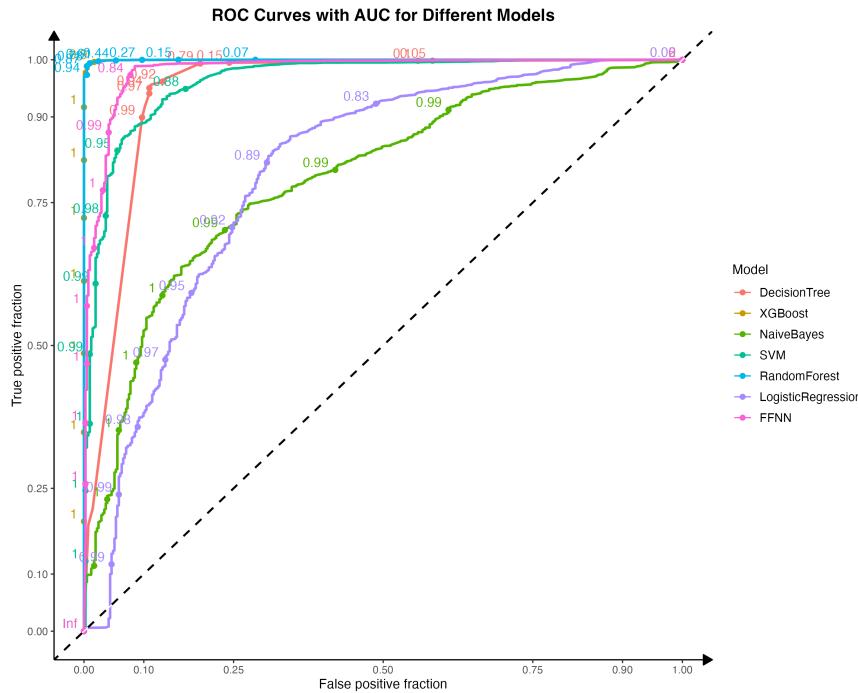
Confusion Matrices



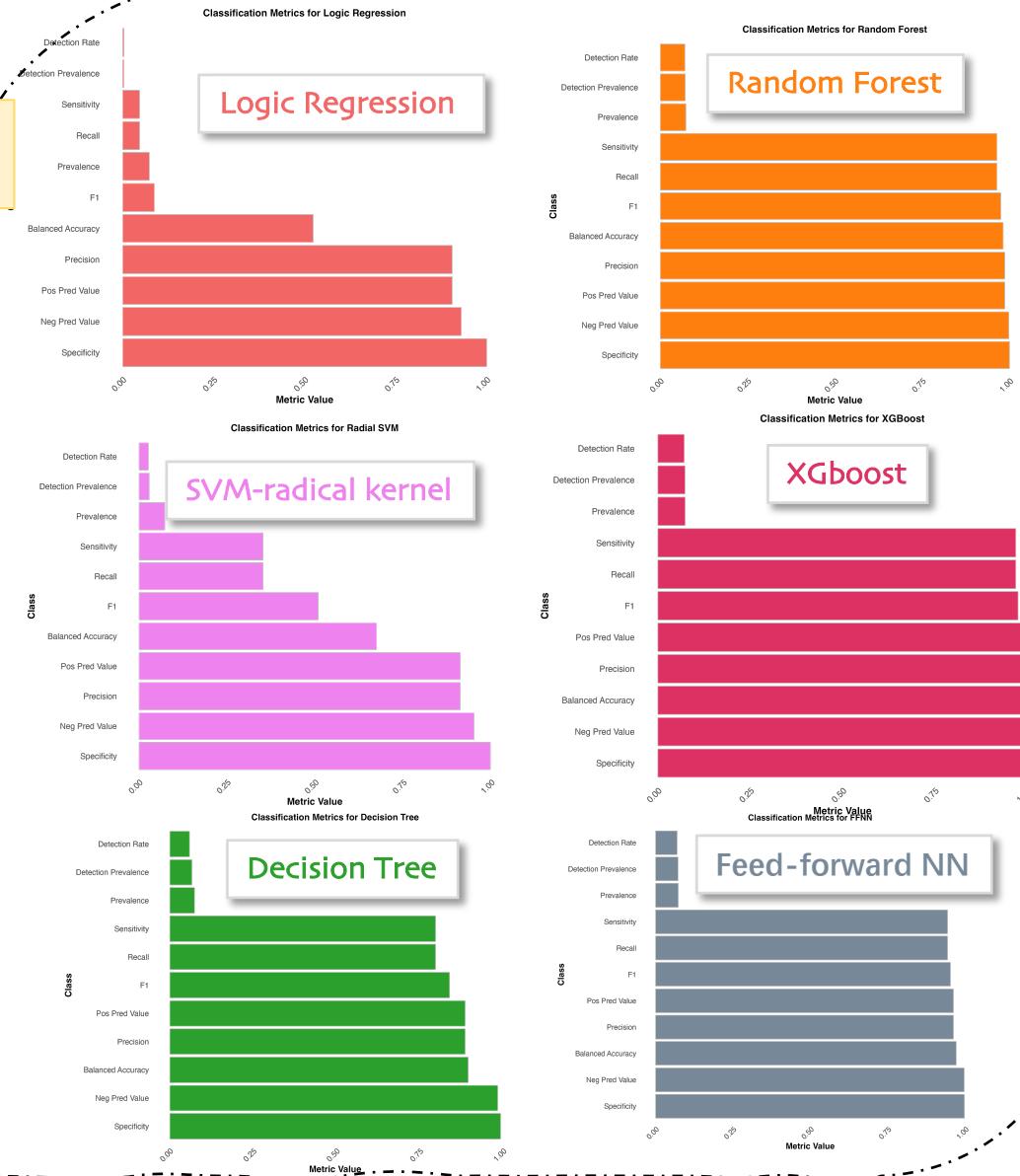
Binary classification – Model Performances

→ **ROCs and metrics** used to evaluate models

ROC-AUCs



Models Metrics



Multi-classification

→ Simplifying Disaster Categorization

- ◆ **Reason 1:** Different countries may use varied terms for the same type of disaster.
- ◆ **Reason 2:** Insufficient data in some categories can hinder model performance.

→ Oversampling technique applied (**SMOTE**)

- ◆ **Reason:** extremely unbalanced in some classes

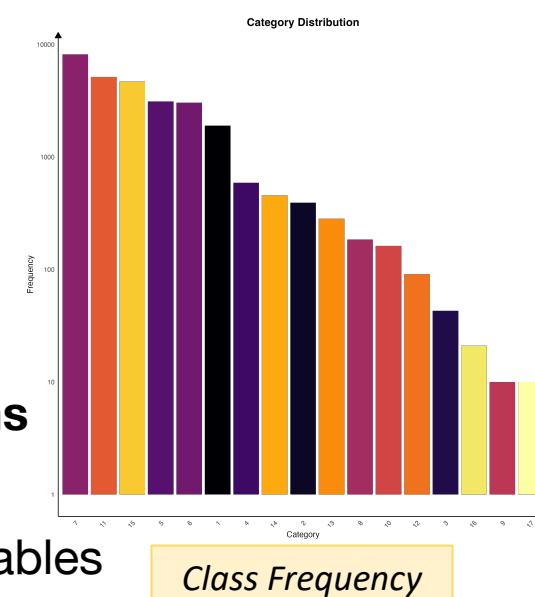
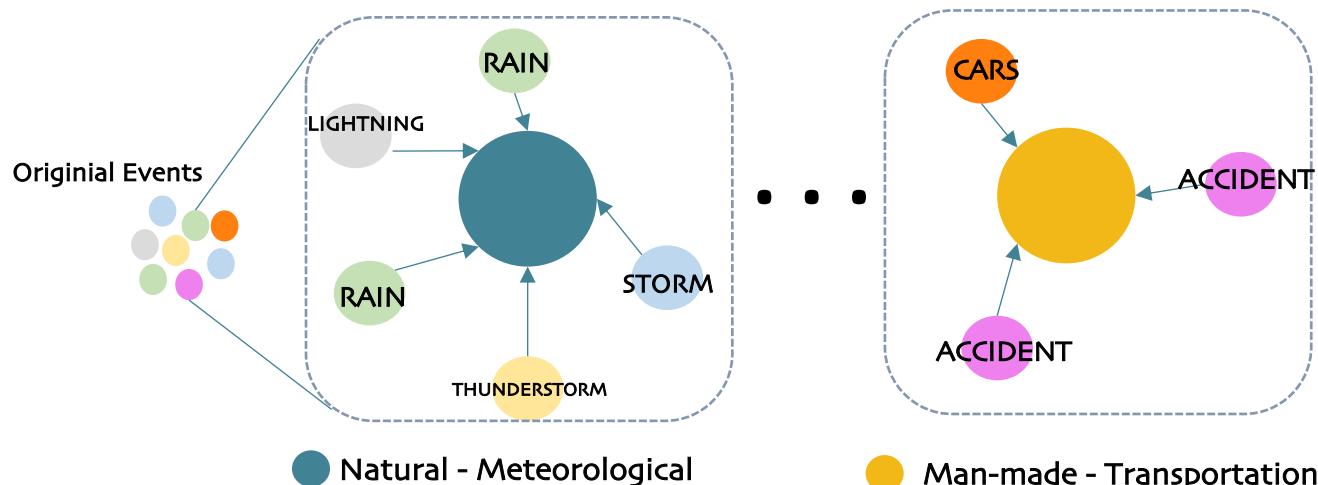
→ Various algorithms are applied

- | | |
|-------------------|-----------------|
| ◆ Multi nom-logic | ◆ FNN |
| ◆ Random Forest | ◆ KNN |
| ◆ SVM - radical | ◆ Decision tree |
| | ◆ XGBoost |

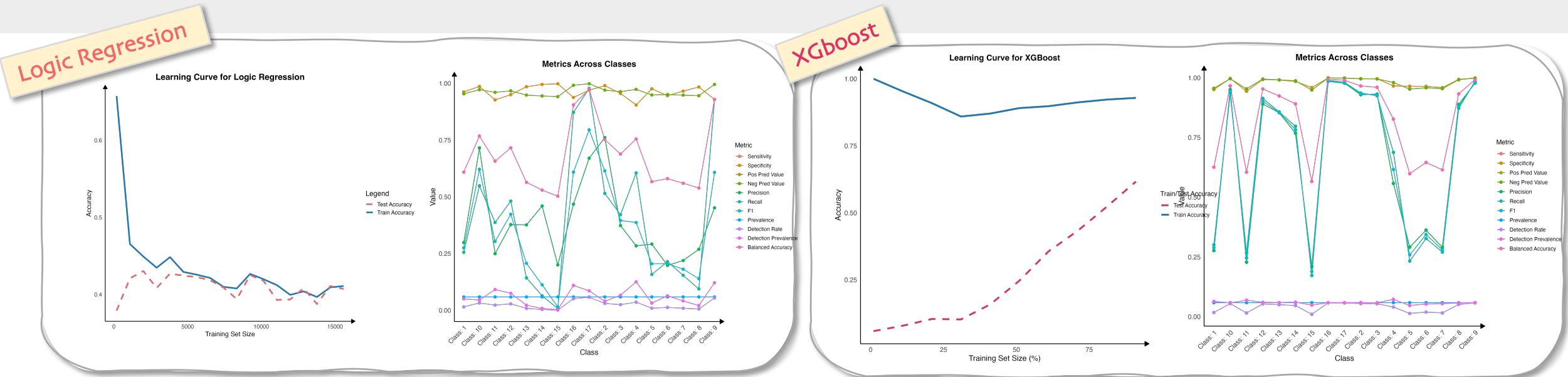
→ Grid searches are applied for top-performing algorithms

(back up slides)

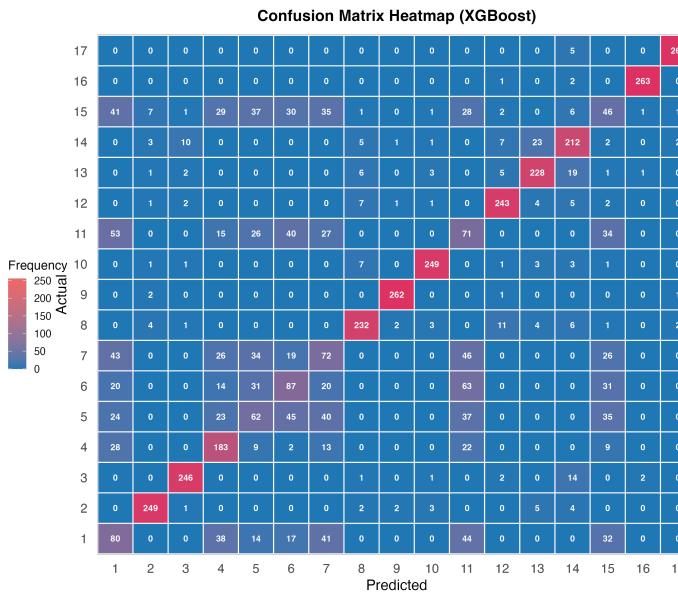
→ PCA loading plots can reveal relationship with original variables



Multi-classification



		Confusion Matrix Heatmap (Logic Regression)																
		Confusion Matrix Heatmap (XGBoost)																
Actual	Predicted																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
16	0	0	12	0	0	0	0	0	0	0	0	6	0	0	0	0	248	0
15	49	8	3	72	15	28	32	0	1	1	41	3	0	0	0	6	6	1
14	0	13	50	0	0	0	0	2	62	1	0	59	6	9	2	36	26	0
13	0	8	37	0	0	0	0	9	40	21	0	41	34	9	0	0	19	48
12	0	0	7	0	0	0	0	7	41	13	0	152	10	4	0	0	28	4
11	45	0	0	37	19	37	25	0	0	0	100	0	0	0	3	0	0	0
10	0	32	15	0	0	0	0	26	5	147	0	10	2	0	0	0	16	13
9	0	0	0	0	0	0	0	0	230	0	0	19	0	0	0	0	0	17
8	0	14	22	0	0	0	0	19	75	6	0	59	1	0	3	53	14	0
7	37	0	0	59	27	35	36	0	0	0	62	0	0	0	10	0	0	0
6	26	0	0	34	32	79	16	0	0	0	74	0	0	0	5	0	0	0
5	35	0	0	65	66	39	29	0	0	0	28	0	0	0	3	0	0	1
4	54	0	0	149	10	9	10	0	0	0	24	0	0	0	9	0	0	1
3	0	0	90	0	0	0	0	0	12	7	0	21	7	2	0	0	123	4
2	0	142	12	0	0	0	0	11	25	11	0	57	2	0	0	0	0	6
1	80	0	0	35	22	32	19	0	0	0	61	0	0	0	11	0	0	6



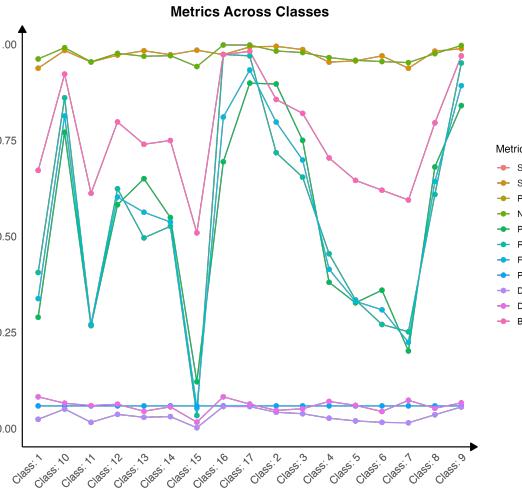
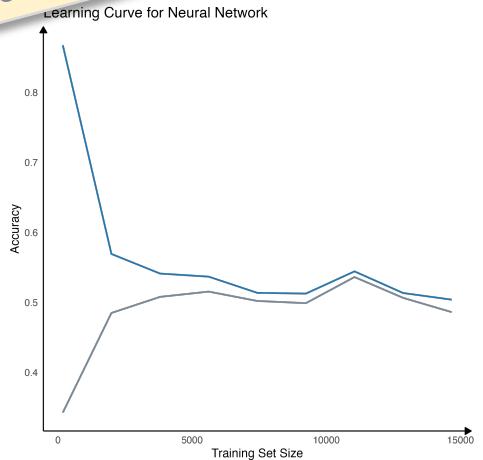
◆ **Logic Regression:**
Good performance in Class **16, 17, 9**

◆ **XGboost:**
Good performance in Class **10, 12, 13, 14, 16, 17, 2, 3, 8, 9**

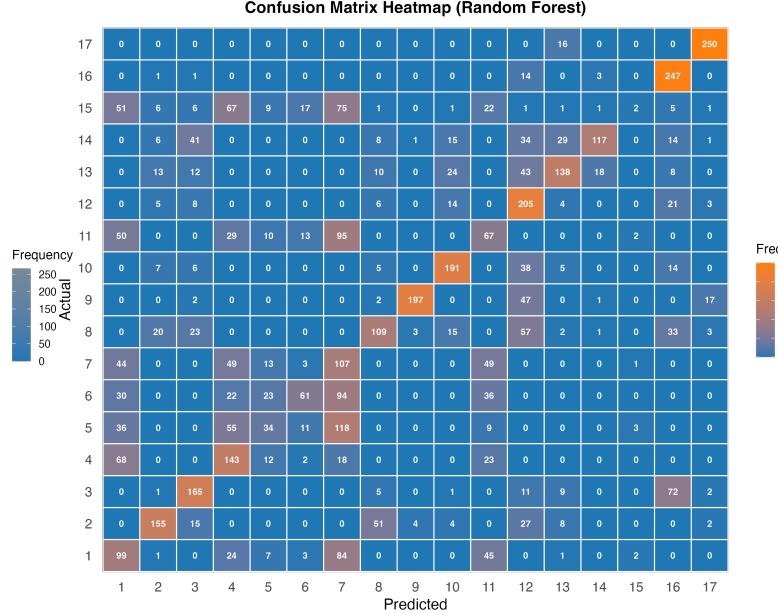
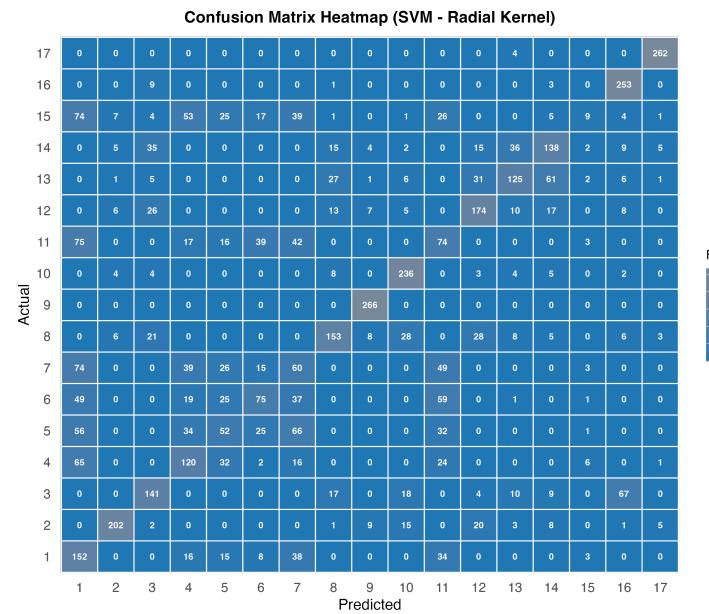
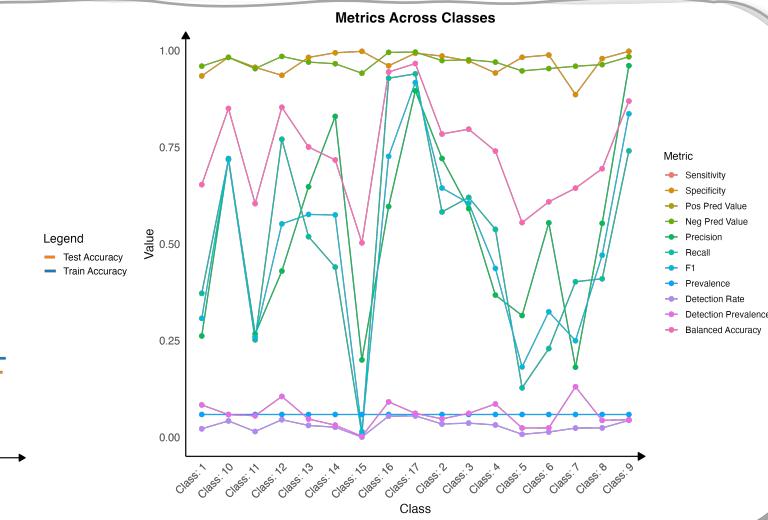
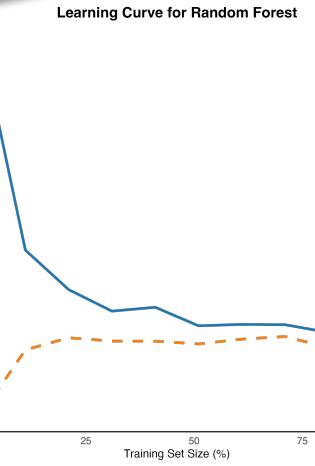
XGboost shows more stable and has higher accuracy and F1 score compared to Logic Regression, but can be improved with more data.

Multi-classification

Feed-forward NN



Random Forest



◆ FNN:

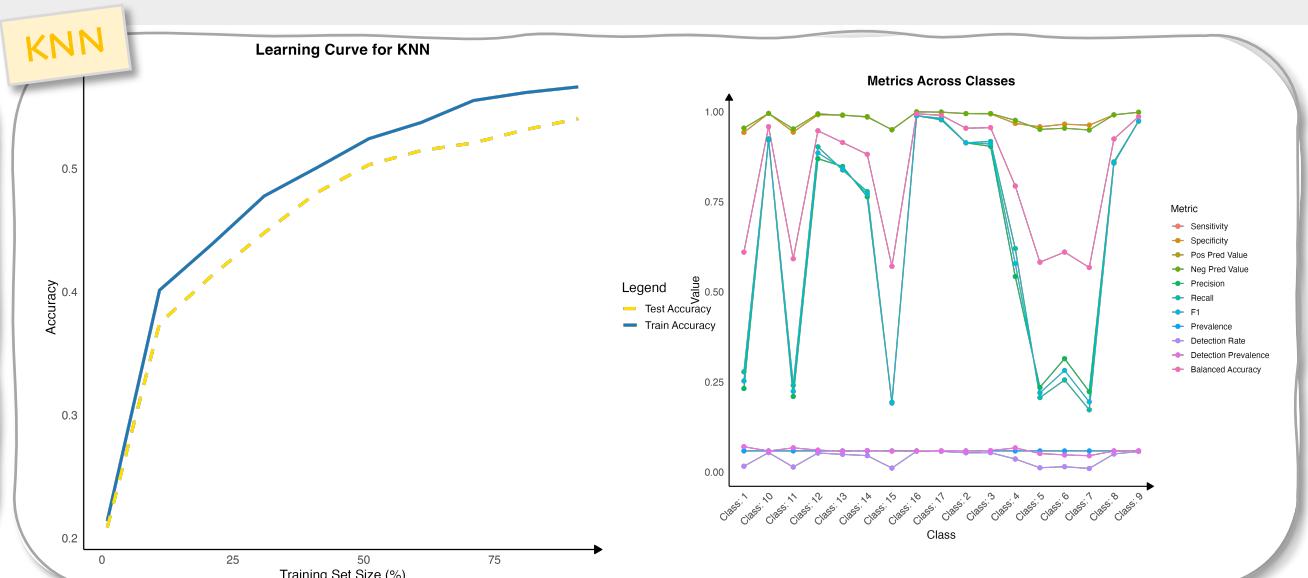
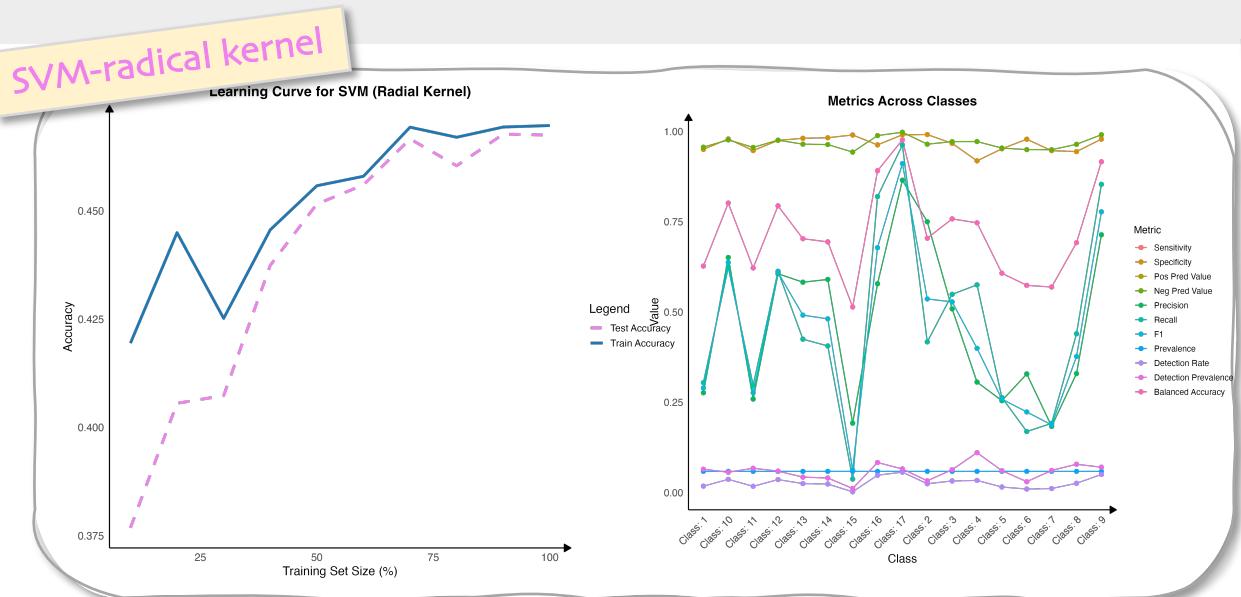
Good performance in Class **10, 16, 17, 9**

◆ Random Forest:

Good performance in Class **16, 17, 9**

◆ All learning curves converged indicating
no over-fitting in two models

Multi-classification



Confusion Matrix Heatmap (SVM - Radial Kernel)																	
Actual	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	250
17	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	250
16	0	1	1	0	0	0	0	0	0	0	0	14	0	3	0	247	0
15	51	6	6	67	0	17	75	1	0	1	22	1	1	1	2	5	1
14	0	8	41	0	0	0	0	8	1	15	0	34	29	117	0	14	1
13	0	13	12	0	0	0	0	10	0	24	0	43	138	18	0	8	0
12	0	5	8	0	0	0	0	6	0	14	0	205	4	0	0	21	3
11	50	0	0	29	10	13	95	0	0	0	67	0	0	0	2	0	0
10	0	7	6	0	0	0	0	5	0	191	0	38	5	0	0	14	0
9	0	0	2	0	0	0	0	2	197	0	0	47	0	1	0	0	17
8	0	20	23	0	0	0	0	109	3	15	0	57	2	1	0	33	3
7	44	0	0	49	13	3	107	0	0	0	49	0	0	0	1	0	0
6	30	0	0	22	23	61	94	0	0	0	36	0	0	0	0	0	0
5	36	0	0	55	34	11	118	0	0	0	9	0	0	0	3	0	0
4	68	0	0	143	12	2	18	0	0	0	23	0	0	0	0	0	0
3	0	1	165	0	0	0	0	5	0	1	0	11	9	0	0	72	2
2	0	155	15	0	0	0	0	51	4	4	0	27	8	0	0	0	2
1	99	1	0	24	7	3	84	0	0	0	45	0	1	0	2	0	0

		Confusion Matrix Heatmap (KNN)																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Actual	17	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	261	
	16	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	263	0
	15	46	6	3	29	35	29	29	1	0	1	26	3	0	6	51	0	1
	14	0	1	15	0	0	0	0	7	1	0	0	3	26	207	2	2	2
	13	0	2	6	0	0	0	0	3	0	5	0	7	223	19	2	0	0
	12	0	3	0	0	0	0	0	8	2	3	0	240	1	5	3	1	0
	11	52	0	0	17	36	28	29	0	0	0	64	0	1	0	39	0	0
	10	0	2	0	0	0	0	0	11	0	245	0	3	2	2	1	0	0
	9	0	4	0	0	0	0	0	1	295	0	0	0	0	0	1	0	1
	8	0	3	2	0	0	0	0	228	1	5	0	12	5	7	1	0	2
	7	56	0	0	28	32	20	46	0	0	0	45	0	0	0	39	0	0
	6	26	0	0	16	38	68	20	0	0	0	63	0	1	1	33	0	0
	5	35	0	0	19	55	47	32	0	0	0	40	0	0	0	38	0	0
	4	30	0	1	165	11	7	17	0	0	0	20	0	0	0	15	0	0
	3	0	2	244	0	0	0	0	1	1	1	0	3	1	13	0	0	0
	2	0	243	0	0	0	0	0	5	2	4	0	4	3	5	0	0	0
	1	74	0	0	30	27	17	33	0	0	0	47	0	0	0	38	0	0

◆ SVM:

Good performance in Class 17, 9

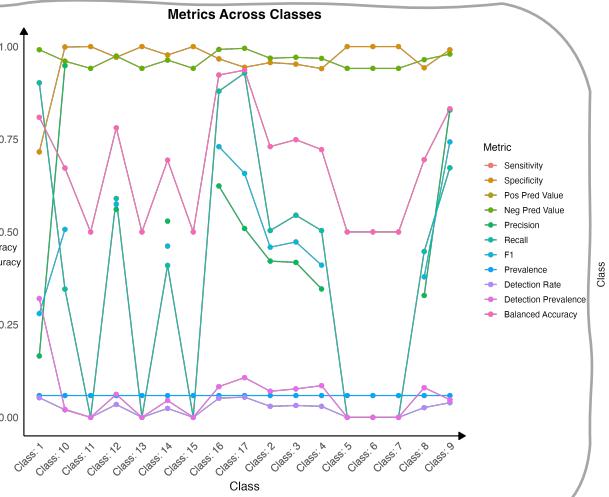
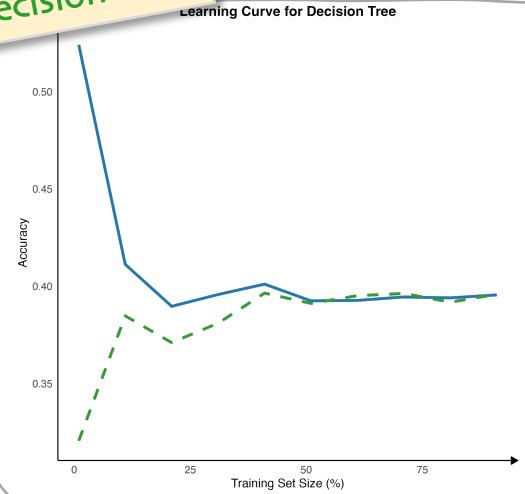
The logo for KNN, featuring a teal diamond shape followed by the letters "KNN:" in a bold, yellow, sans-serif font.

Good performance in Class **10, 12, 13, 14, 16, 17, 2, 3, 8, 9**

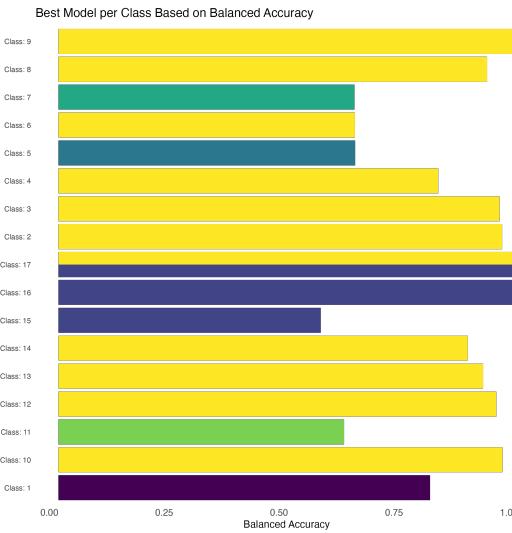
- ◆ KNN is more stable compared to SVM

Multi-classification

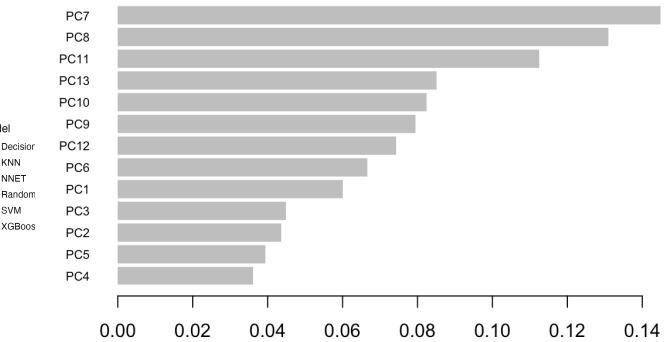
Decision Tree



Highest Accuracy Models



Importance plot



Decision tree confusion matrix

Confusion Matrix Heatmap (Decision Tree)

Actual	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	258	
16	0	0	3	0	0	0	0	0	0	0	0	0	4	0	259	0		
15	51	7	3	53	30	20	53	2	0	2	27	0	0	3	9	5	1	
14	0	4	29	0	0	0	0	18	7	4	0	10	35	140	0	12	7	
13	0	1	2	0	0	0	0	18	1	9	0	30	132	57	0	6	10	
12	0	3	8	0	0	0	0	17	15	10	0	166	16	17	0	14	0	
11	46	0	0	25	27	27	58	0	0	0	72	0	0	0	11	0	0	
10	0	6	6	0	0	0	0	6	0	229	0	7	4	7	0	1	0	
9	0	0	0	0	0	0	0	0	253	0	0	12	0	0	0	0	1	
8	0	1	6	0	0	0	0	0	162	10	23	0	41	3	10	0	7	3
7	48	0	0	38	33	22	67	0	0	0	49	0	0	0	9	0	0	
6	30	0	0	24	40	72	44	0	0	0	49	0	0	0	7	0	0	
5	37	0	0	37	89	23	43	1	0	0	23	0	0	0	13	0	0	
4	53	0	0	121	29	11	27	0	0	0	13	0	0	0	12	0	0	
3	0	0	174	0	0	0	0	2	0	7	0	0	3	10	0	69	1	
2	0	191	1	0	0	0	0	12	15	13	0	19	2	7	0	0	6	
1	108	0	0	20	24	25	40	0	0	0	36	0	0	0	13	0	0	

Summary

- With various algorithms, all classes accuracies can be larger than 50%
- XGBoost and KNN shows better performances
- XGboost is time consuming and possible overfitted
- Different classes are suitable for different algorithms

Summary

- XGBoost is a good choice for class 2,3,4,6,8,9,10,12,14,17
- KNN performs well for class 15, 16, 17
- SVM works best for class 11
- Random Forest is suitable for class 7
- FNN is suitable for class 5
- Decision Tree is suitable for class 1