Effective Machine Learning Based Format Selection and Performance Modeling for SpMV on GPUs

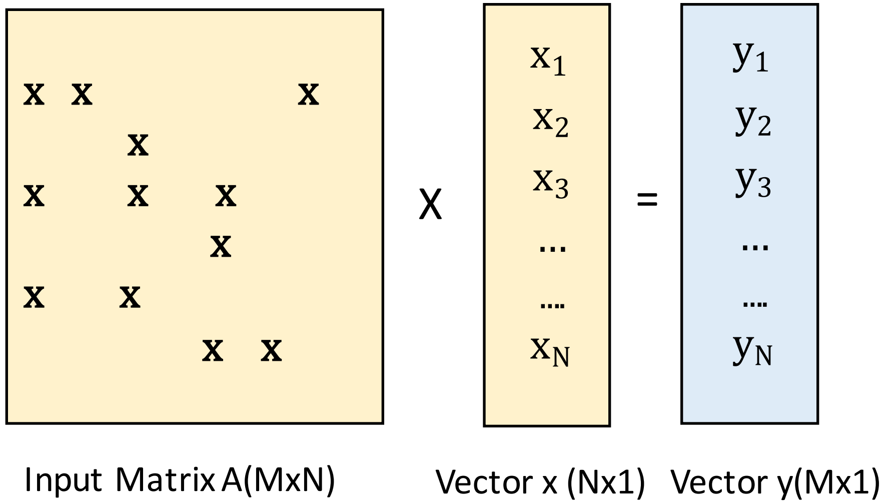
Israt Nisa[*], Charles Siegel[+], Aravind Sukumaran Rajam[*], Abhinav Vishnu[+], P. Sadayappan[*]

[*]The Ohio State University

[+]Pacific Northwest National Laboratory

Sparse Matrix-Vector Multiplication

$$Ax = y$$



Applications of SpMV

Linear Algebra

Graph Analytics

Fluid Dynamics

Deep Learning

eigenvalue problems

Recent Formats of SpMV

Yet, no absolute winner!

yaSpMV-2014
Yan et al.

lightspmv-2015
Liu et al.

CSR5-2015
Liu et al.

clSpMV-2012
Su et al.

Adaptive-2016
Zardoshti et al.

merge-spmv-2012
Duan et al.

HolaSpMV-2017
Steinberger et al.

Adaptive-CSR-2015
Mayank et al.

CSR, ELL-2009
Bell et al.

Recent Works on Format Selection and Performance Modeling

Classification

- Decision Tree – Li et al. (PLDI-2013), Sedaghati et al. (ICS 2015)
- Support Vector Machine (SVM)– Benatial et al. (ICPP 2016)
- Deep learning – Zhao et al. (PPoPP 2018), Cui et al. (MCSoC 2016)

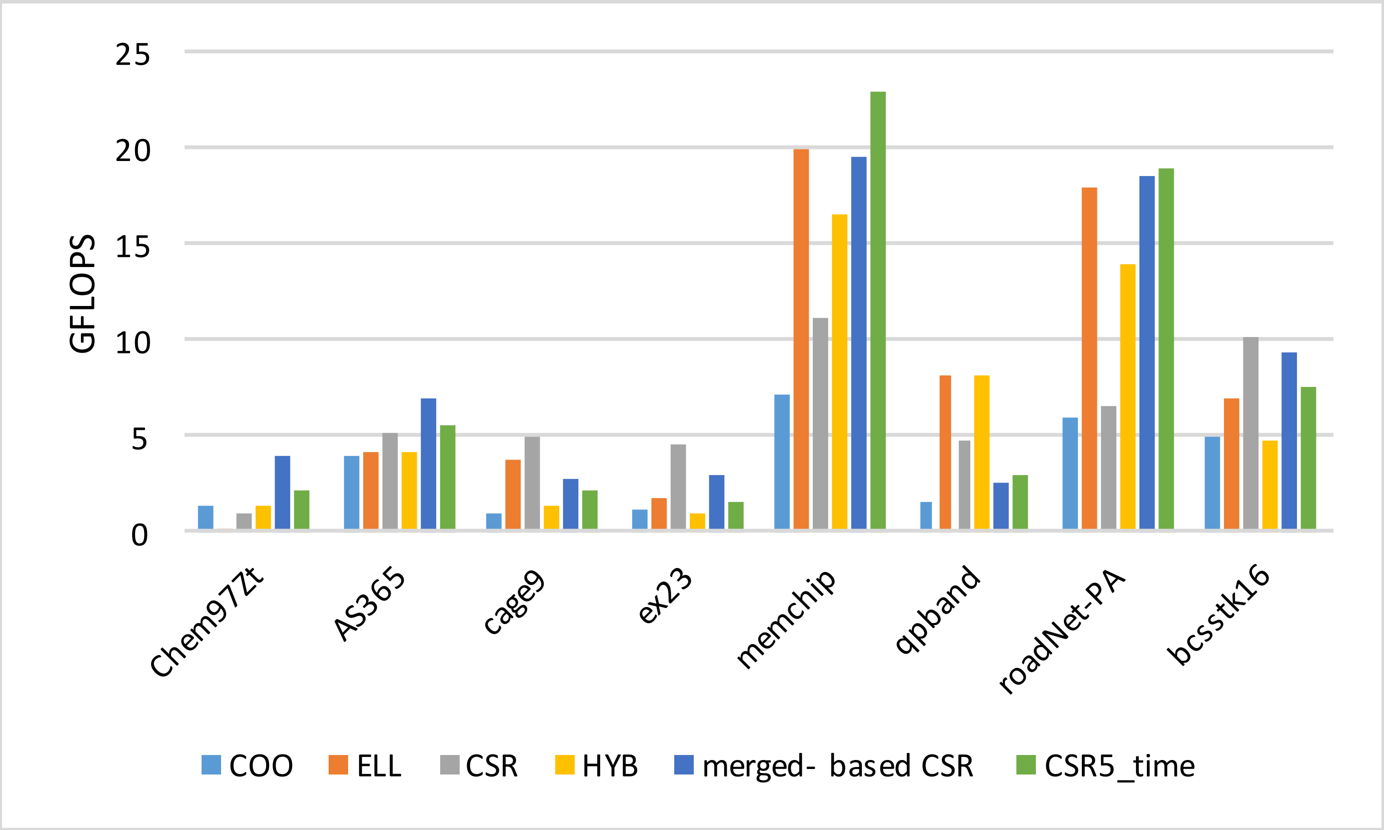
Performance modeling

- Analytical modeling – Zhao et al. (HPCA 2011), Zardoshti et al. (J-SC 2016), Guo et al. (CC 2015)
- Multi Layer Perceptron (MLP) – Benatia et al. (ICPADS 2016)
- Support Vector Regression (SVR) – Benatia et al. (ICPADS 2016)

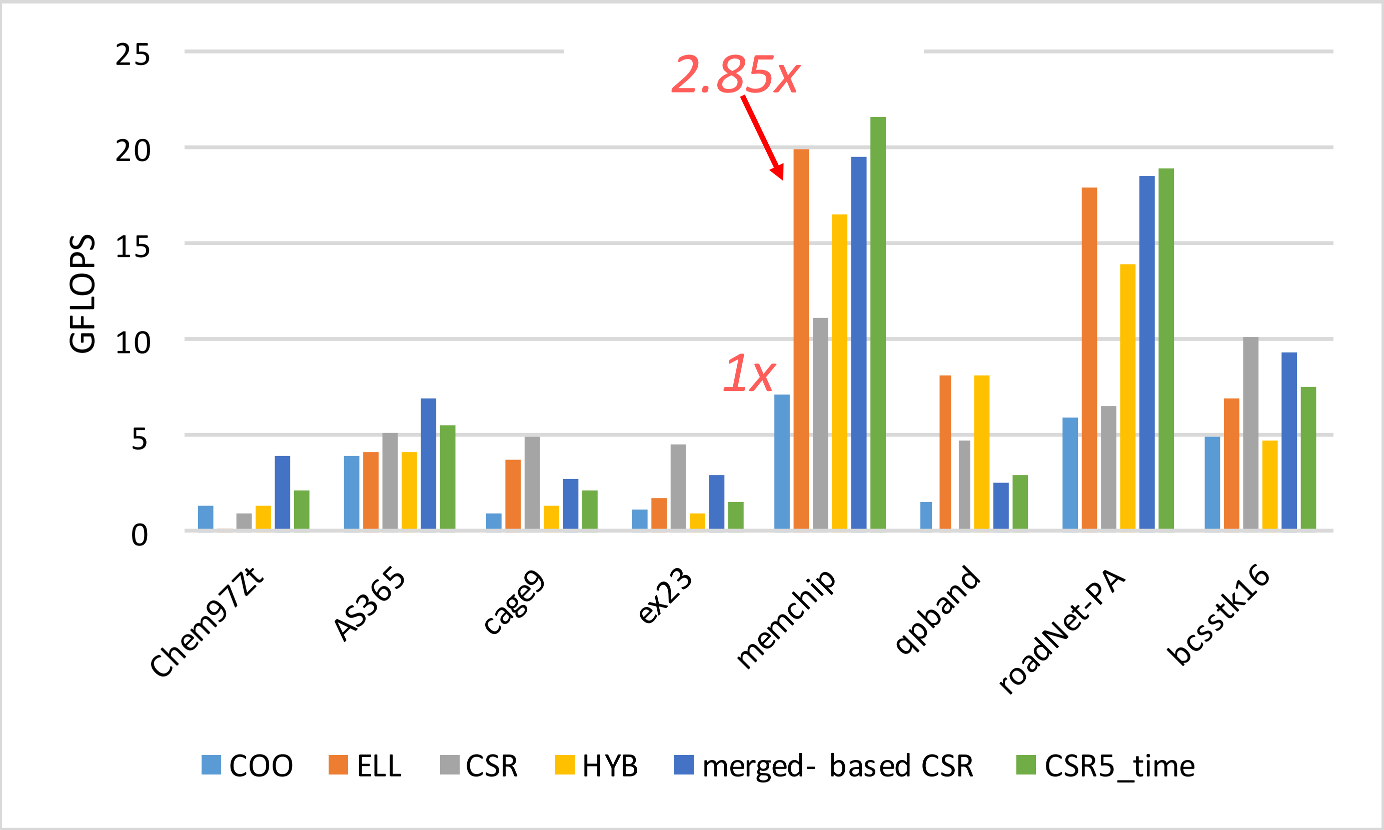
Problems addressed

1. A model to efficiently predict the best-performing format for a unseen sparse matrix for GPU
2. Can the SpMV execution time for an unseen sparse matrix be effectively predicted for various representation formats?

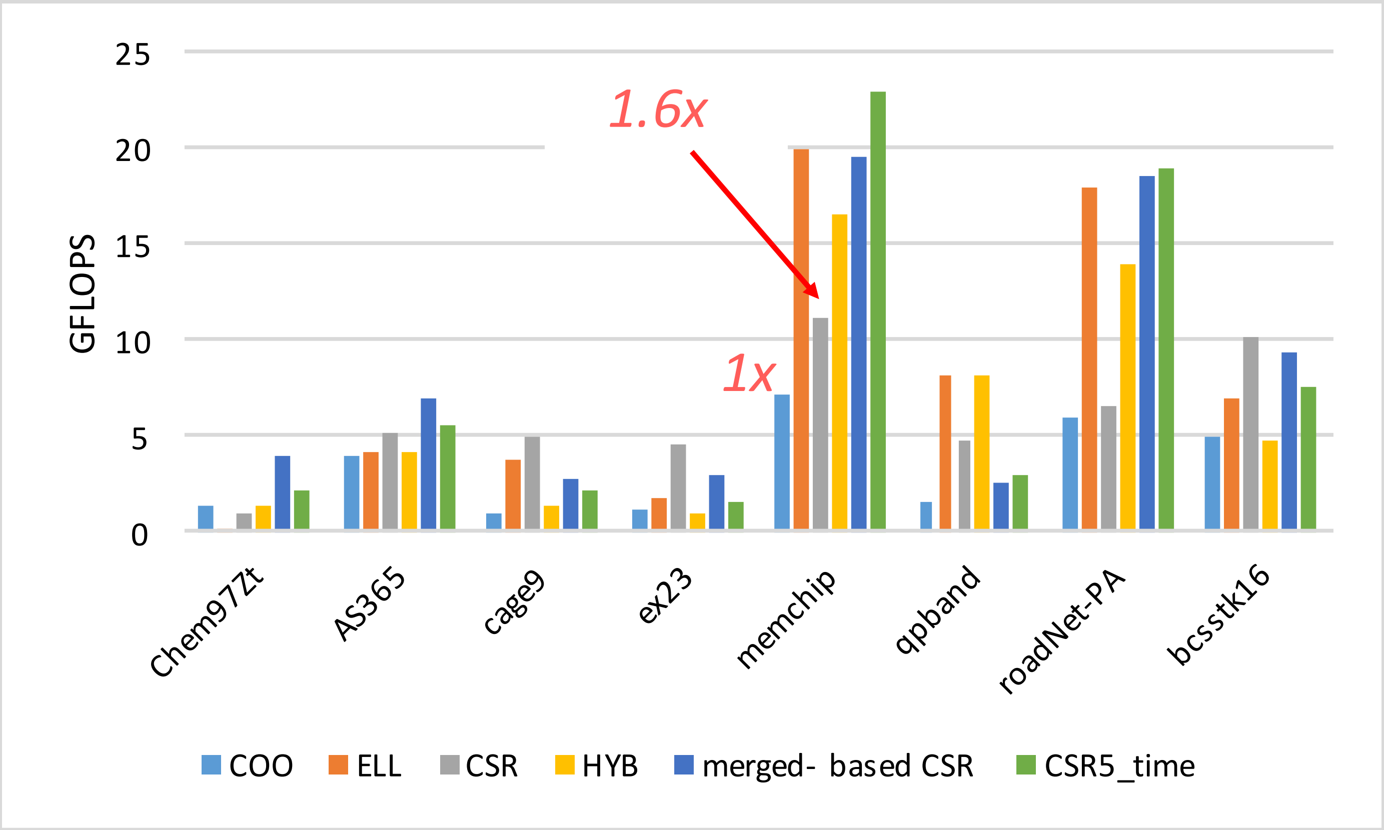
Performance Variation across formats on GPU P100



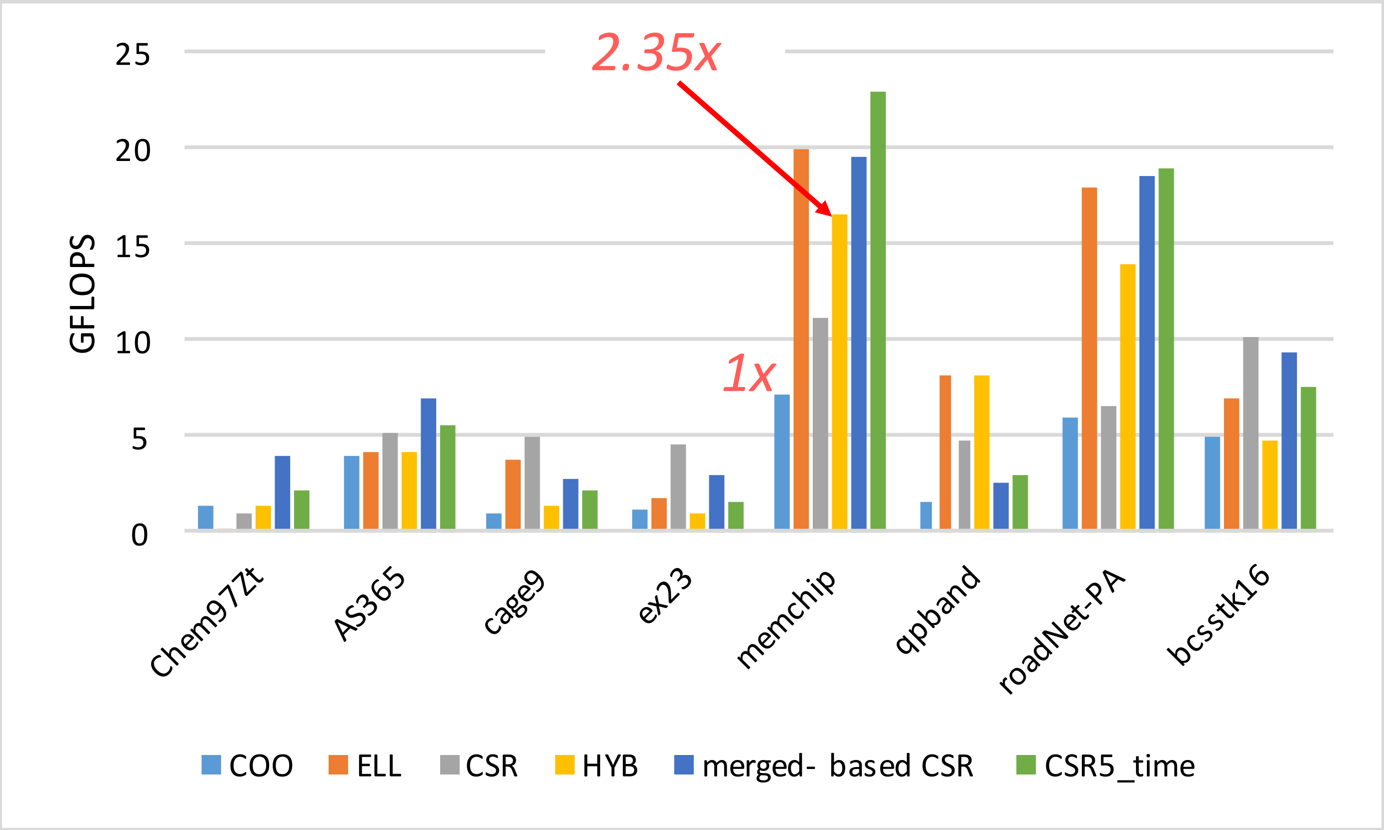
Performance Variation across formats



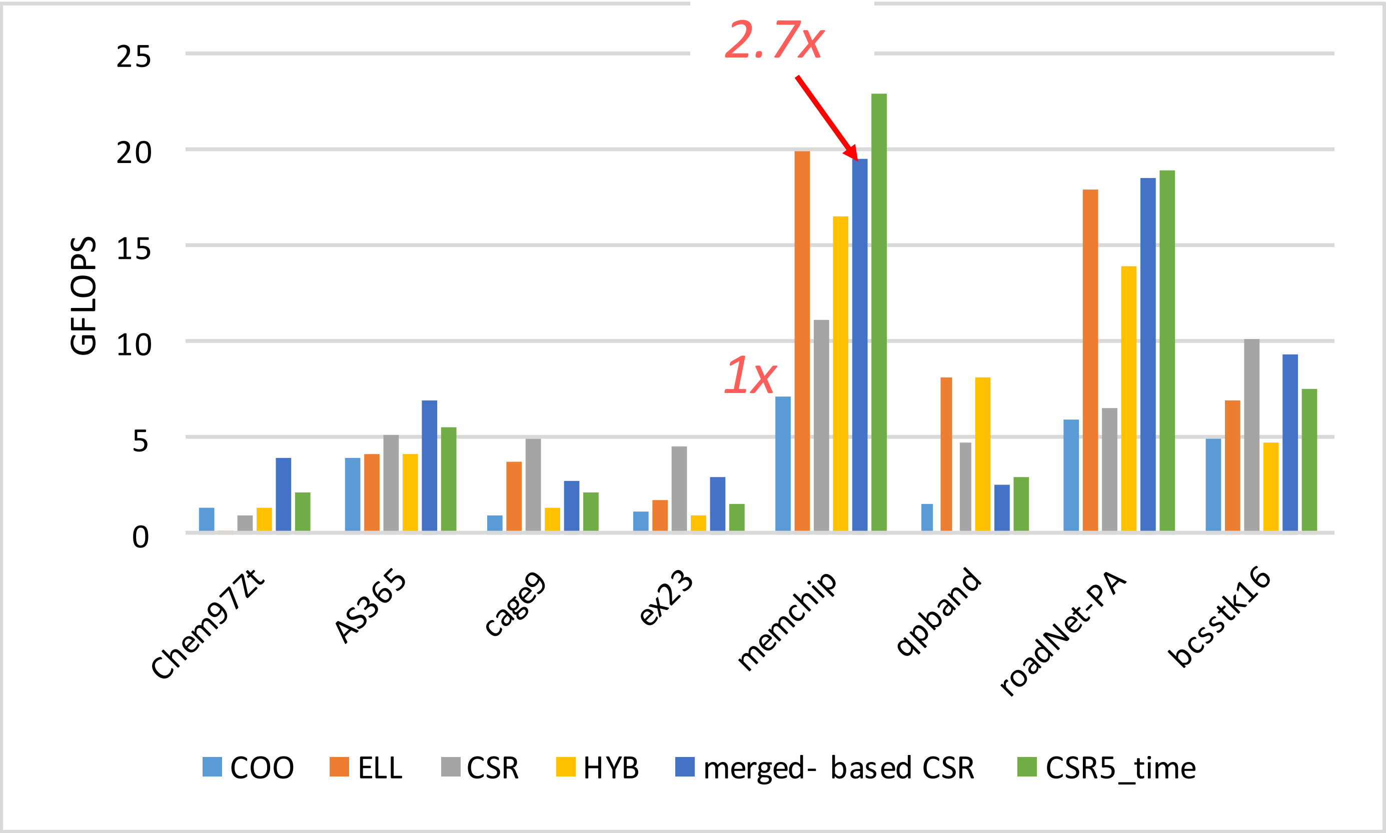
Performance Variation across formats



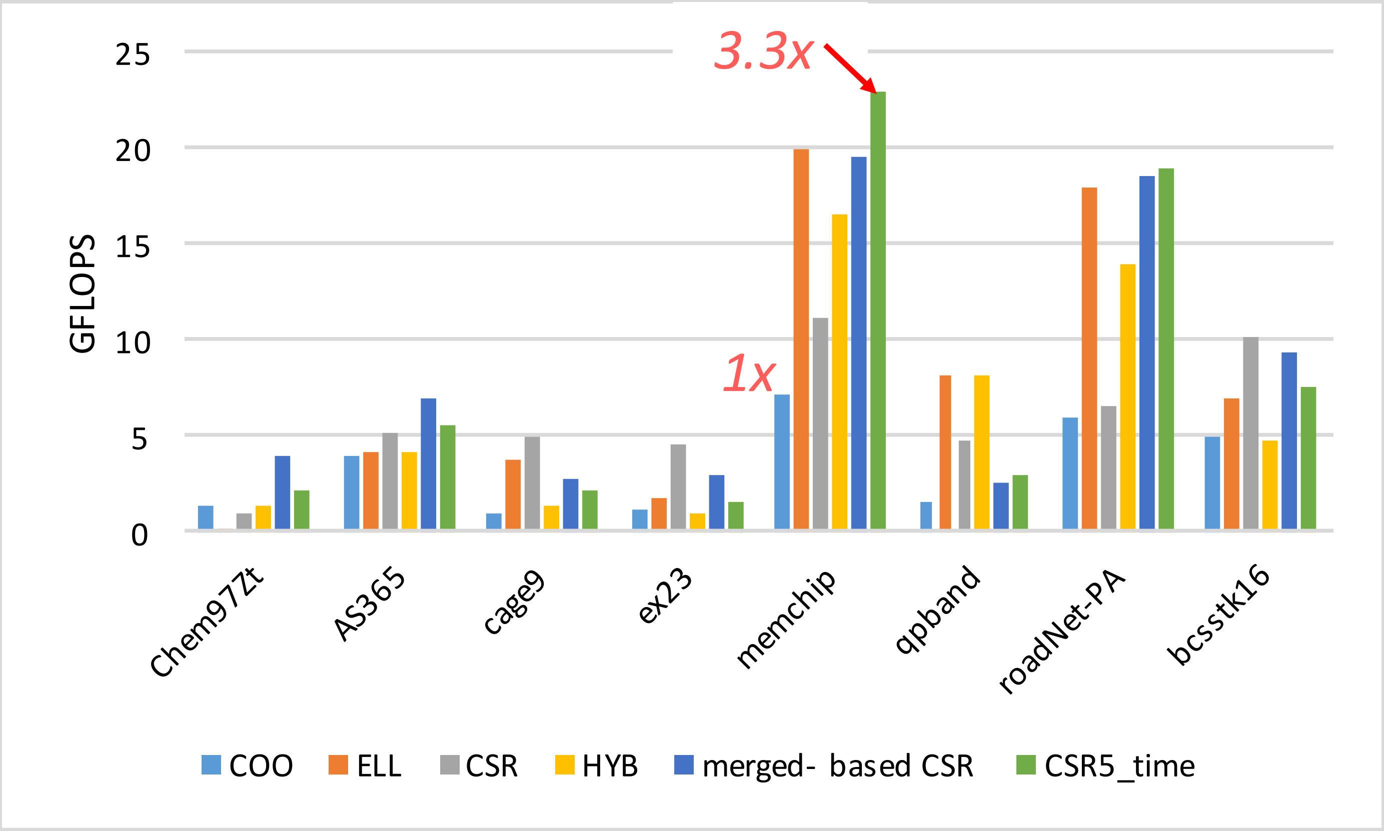
Performance Variation across formats



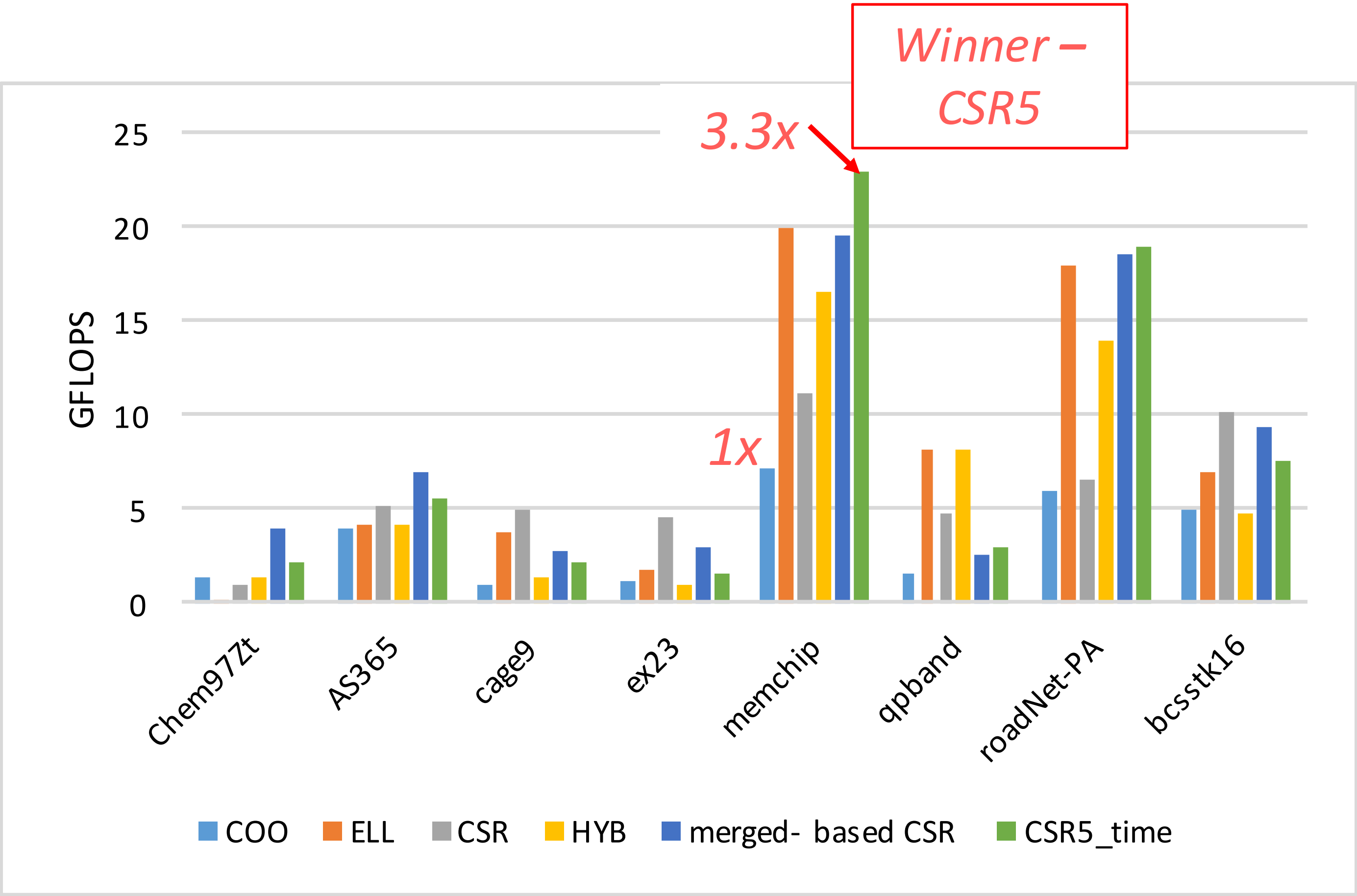
Performance Variation across formats



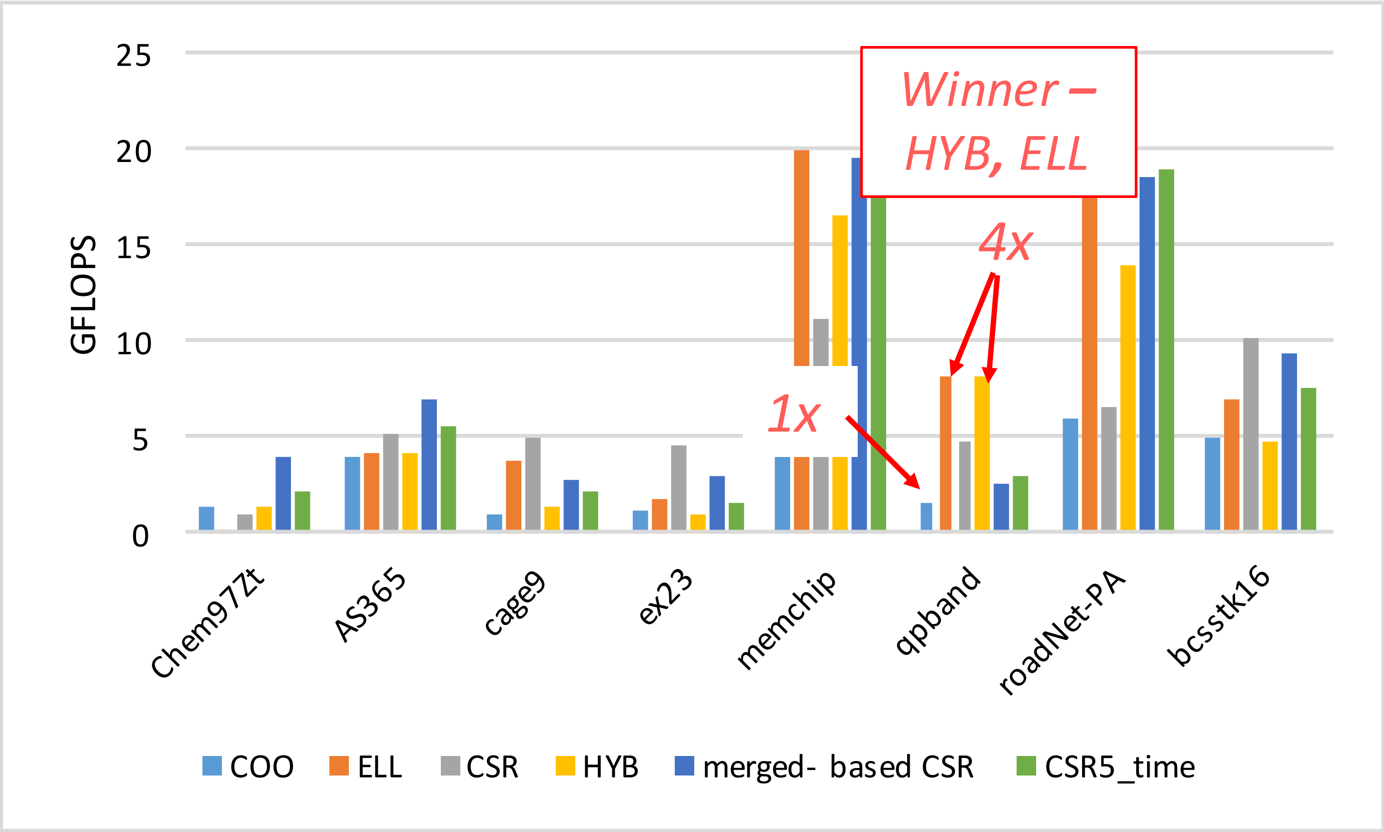
Performance Variation across formats



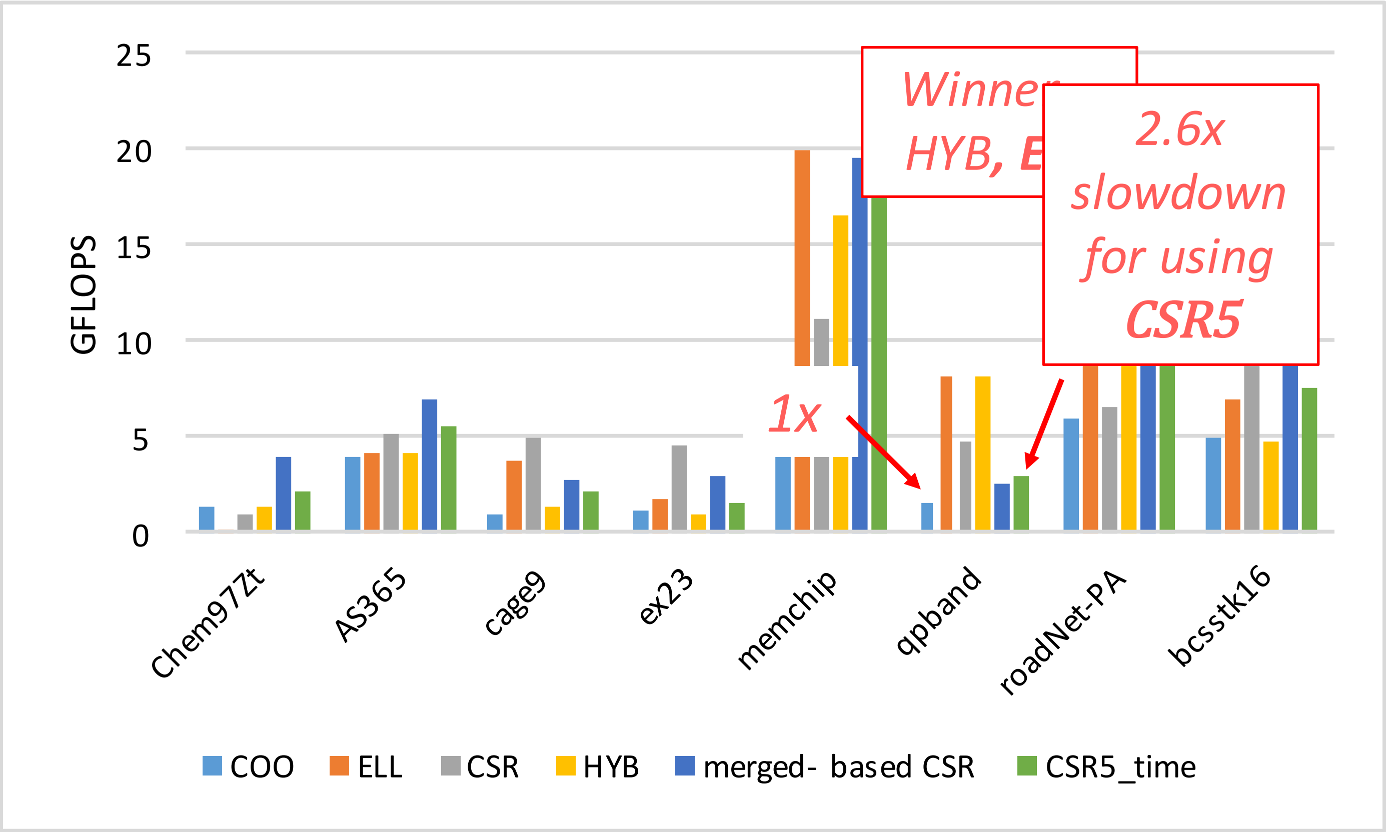
Performance Variation across formats



Performance Variation across formats



Performance Variation across formats



How about all matrices from the Florida repository?

	Avg. slowdown	>2x slowdown
COO	3.37	2077
ELL	12.43	1154
CSR	2.29	362
HYB	3.28	1521
CSR5	1.60	362
merged CSR	1.42	104

Can we use 1 format for all matrices?

	Avg. slowdown	>2x slowdown
COO	3.37	2077
ELL	12.43	1154
CSR	2.29	362
HYB	3.28	1521
CSR5	1.60	362
merged CSR	1.42	104

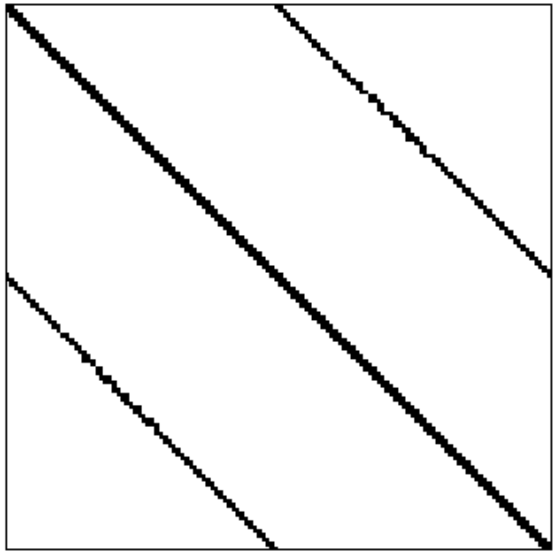
Can we use 1 format for all matrices?

	Avg. slowdown	>2x slowdown
COO	3.37	2077
ELL	12.43	1154
CSR	2.29	362
HYB	3.28	1521
CSR5	1.60	362
merged CSR	1.42	104

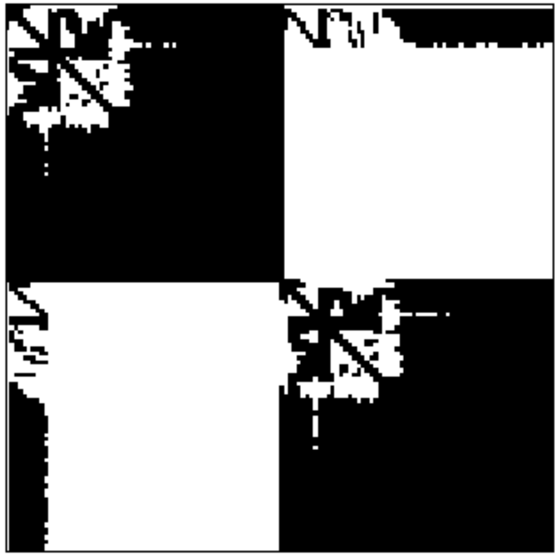
Up to 4x slowdown

Performance Variation in Advanced formats!

matrix	n_rows	n_cols	nnz_tot	CSR5_flops	mergeCSR flops
rgg_n_2_19_s0	524,288	524,288	6,539,532	22	21
auto	448,695	448,695	6,629,222	18	15



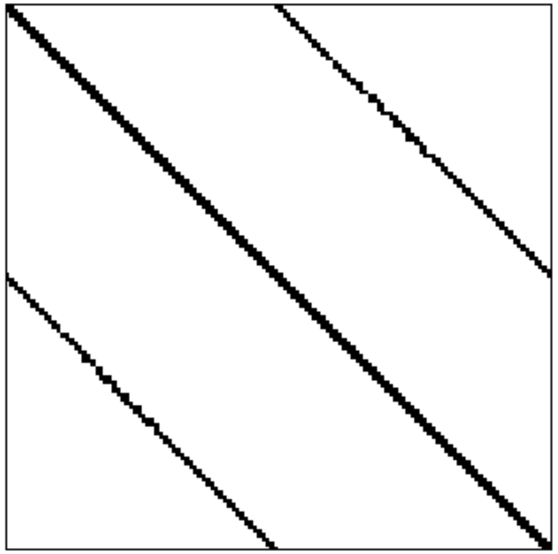
rgg_n_2_19_s0



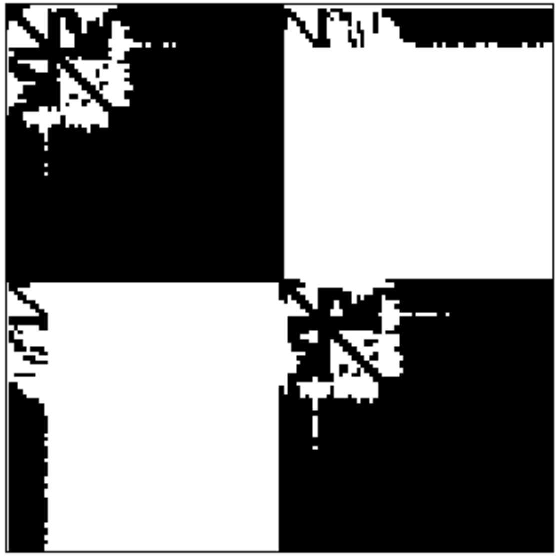
auto

Performance Variation in Advanced formats!

matrix	n_rows	n_cols	nnz_tot	CSR5_flops	mergeCSR flops
rgg_n_2_19_s0	524,288	524,288	6,539,532	22	21
auto	448,695	448,695	6,629,222	18	15



rgg_n_2_19_s0



auto

GFLOPS is Not a function of nnz!

Sparse Matrix Storage Formats

	0	1	2	3
0	a	0	b	0
1	0	0	c	0
2	d	e	f	g
3	0	0	h	0

Sparse matrix

Sparse Matrix Storage Formats

	0	1	2	3
0	a	0	b	0
1	0	0	c	0
2	d	e	f	g
3	0	0	h	0

Sparse matrix

row_ind	0	0	1	2	2	2	2	3
col_ind	0	2	2	0	1	2	3	2
val	a	b	c	d	e	f	g	h

a) COO representation

row_ptr	0	2	3	7	8			
col_ind	0	2	2	0	1	2	3	2
val	a	b	c	d	e	f	g	h

b) CSR representation

Sparse Matrix Storage Formats

	0	1	2	3
0	a	0	b	0
1	0	0	c	0
2	d	e	f	g
3	0	0	h	0

Sparse matrix

col_ind			
0	2	0	0
2	0	0	0
0	1	2	3
2	0	0	0

val			
a	b	0	0
c	0	0	0
d	e	f	g
h	0	0	0

c) ELL representation

Tile 1:

col_ind	
0	2
2	0

val	
a	c
b	d

w, s

Tile 2:

col_ind	
1	3
2	2

val	
e	g
f	h

d) CSR5 representation (w=2, s=2)

SpMV Format Selection Problem

Matrix Features

a_1	a_2	a_3	a_4	a_5

x	x	x		
		x		x
				x
	x	x	x	x
		x		x
x		x		

Matrix Features – Feature set 1

a_1	a_2	a_3	a_4	a_5
x	x	x		
		x		x
				x
	x	x	x	x
		x		x
x		x		

rows = 6

Matrix Features – Feature set 1

a_1	a_2	a_3	a_4	a_5
x	x	x		
		x		x
				x
	x	x	x	x
		x		x
x		x		

cols = 5

Matrix Features – Feature set 1

nnz = 14
nnz_mu = 2.3
density = .58

a_1	a_2	a_3	a_4	a_5
x	x	x		
		x		x
				x
	x	x	x	x
		x		x
x		x		

Matrix Features – Feature set 1

Complexity
$O(1)$

nnz
nnz_mu
density

a_1	a_2	a_3	a_4	a_5
x	x	x		
		x		x
				x
	x	x	x	x
		x		x
x		x		

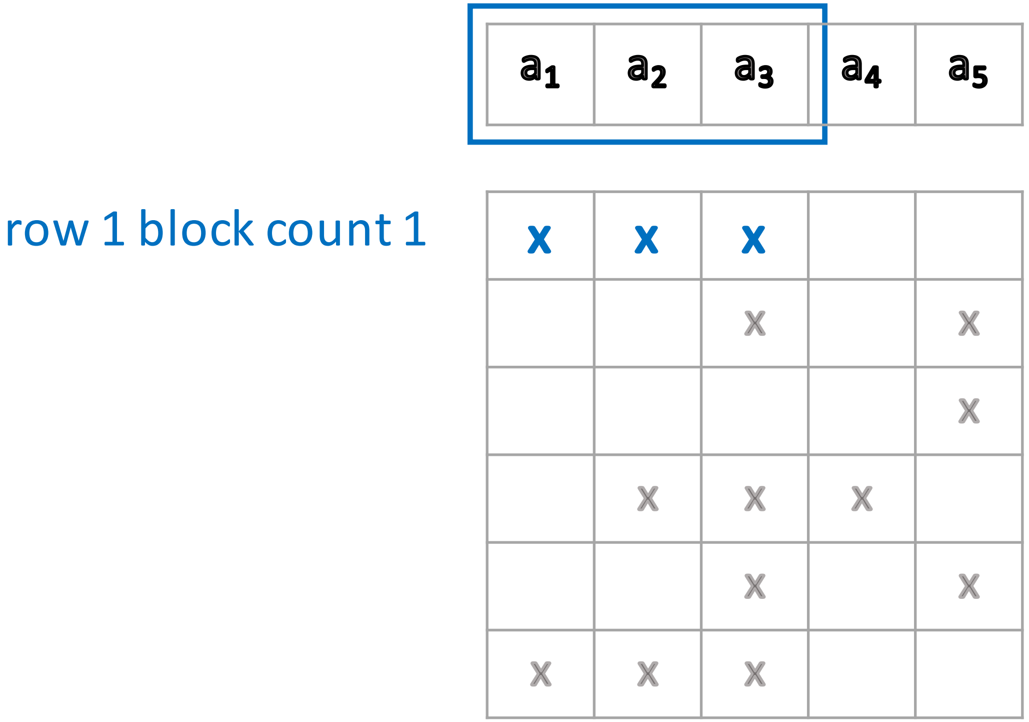
Matrix Features – Feature set 2

Complexity
$O(nnz)$

nnz_max = 4
nnz_sigma = .95

a_1	a_2	a_3	a_4	a_5
x	x	x		
		x		x
				x
	x	x	x	x
		x		x
x	x	x		

Matrix Features – Feature set 3



Matrix Features – Feature set 3

row 2 block count 2

a_1	a_2	a_3	a_4	a_5
x	x	x		
		x		x
				x
	x	x	x	
		x		x
x	x	x		

Matrix Features – Feature set 3

a_1	a_2	a_3	a_4	a_5

row 3 block count 1

x	x	x		
		x		x
				x
	x	x	x	
		x		x
x	x	x		

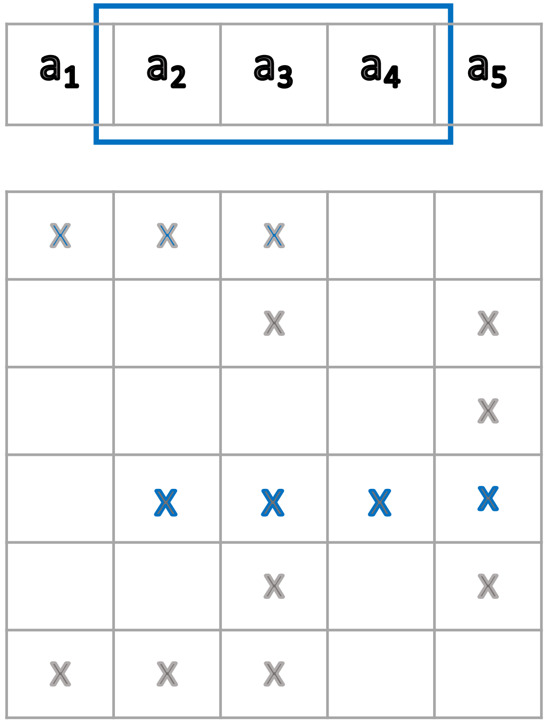
Matrix Features – Feature set 3

Complexity
$O(nnz)$

row 1 block count 1

…

…



Matrix Features

set	feature	description
1	rows, cols	number of rows and columns
	nnz	number of non zero elements
	nnz_mu	average nnz per row
	density	density of the matrix
2	nnz_max	maximum number of nnz in a row
	nnz_sigma	standard dev. of nnz per row
	row_block_count_*	avg. and std. deviation of the number of continuous nnz chunk per row
	row_block_size_*	avg. and std. deviation of the size of continuous nnz chunks in a row
3	block_count	total number of the continuous nnz chunks
	row_block_count_*	min and max of the number of continuous nnz chunks in a row
	row_block_size_*	min and max of the size of continuous nnz chunks in a row

Machine Learning Models

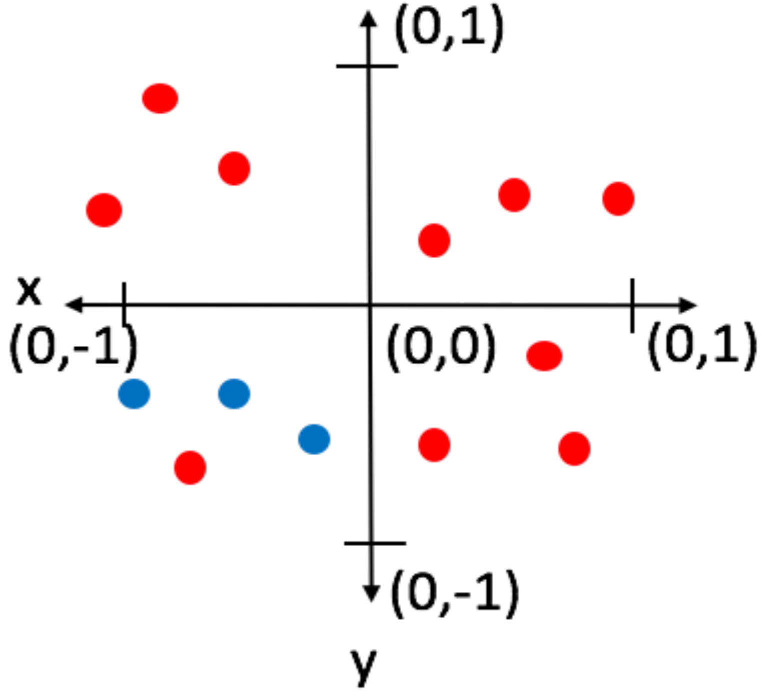
Base models:

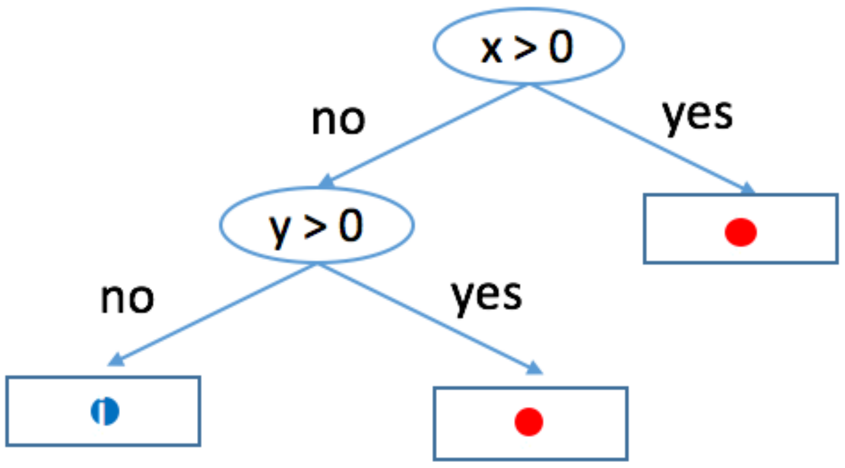
- Decision Tree (D. Tree)
- Support Vector Machine (SVM)
- Multi-layer Perceptron (MLP)

Ensemble models:

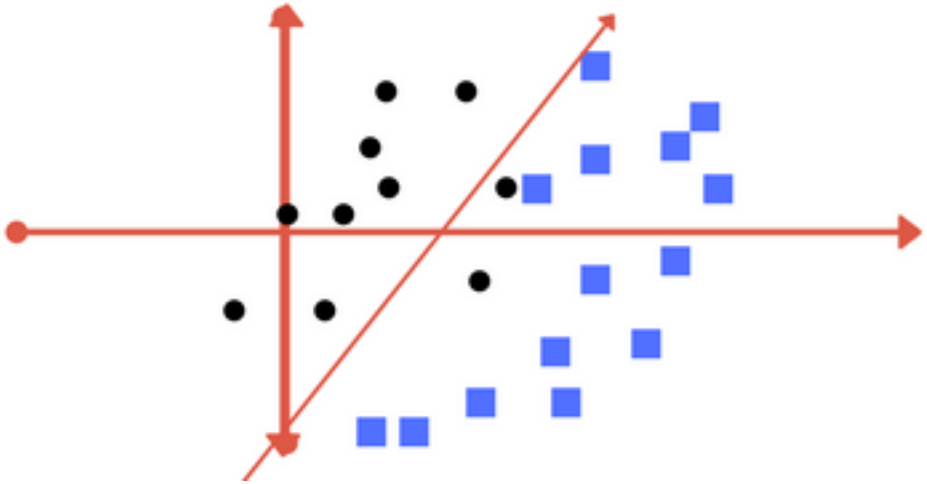
- Gradient Boosted Decision Tree (XGBoost)
- MLP – Ensemble (MLP ens.)

Machine Learning Models – Decs. Tree



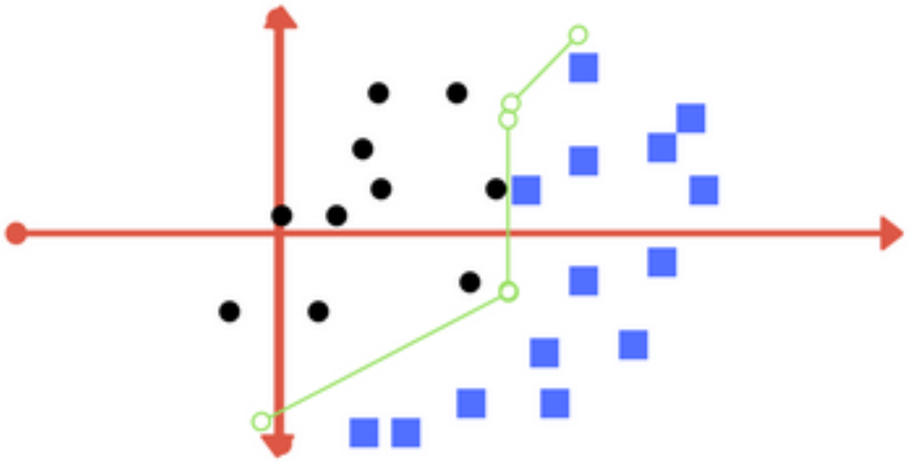


Machine Learning Models - SVM



Source from: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

Machine Learning Models - SVM

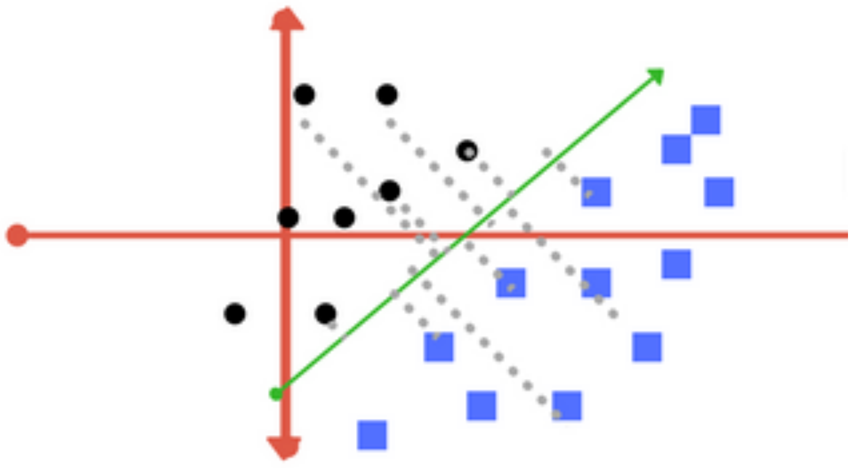


High regularization parameter

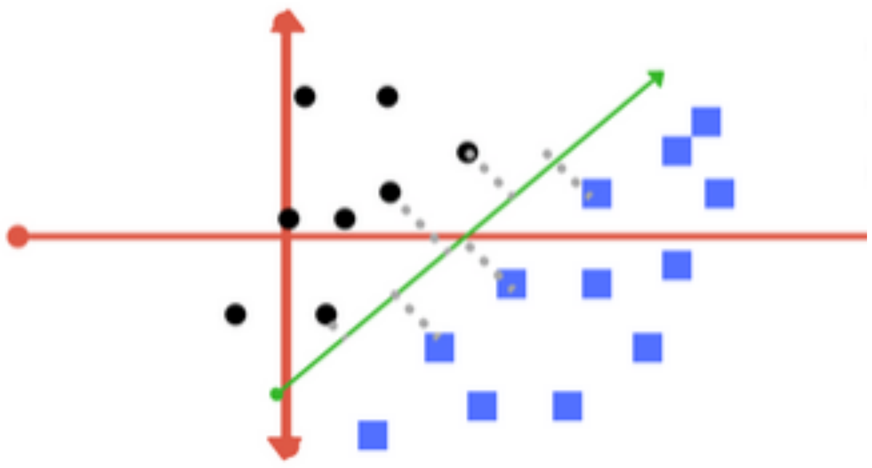
Source from: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

Machine Learning Models - SVM

Low Gamma

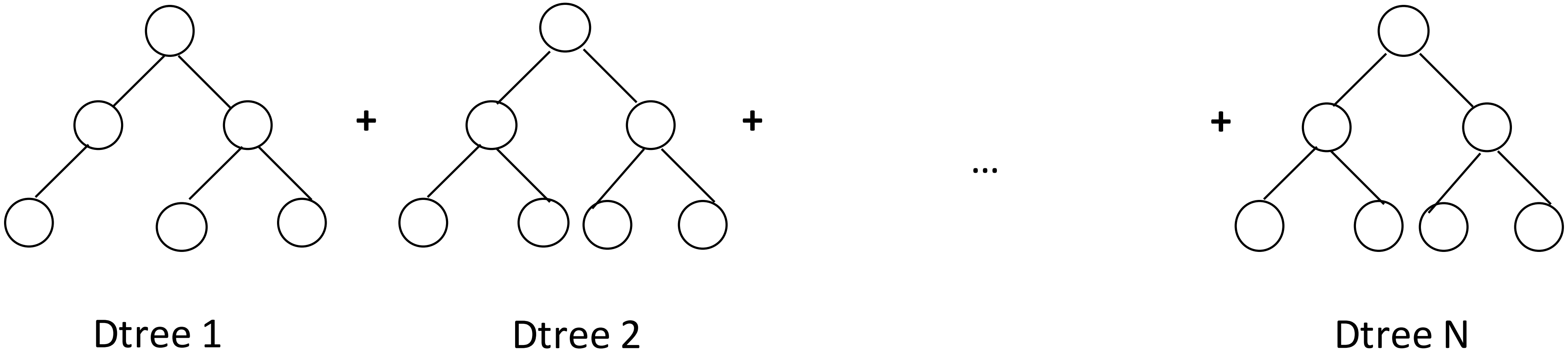


High Gamma



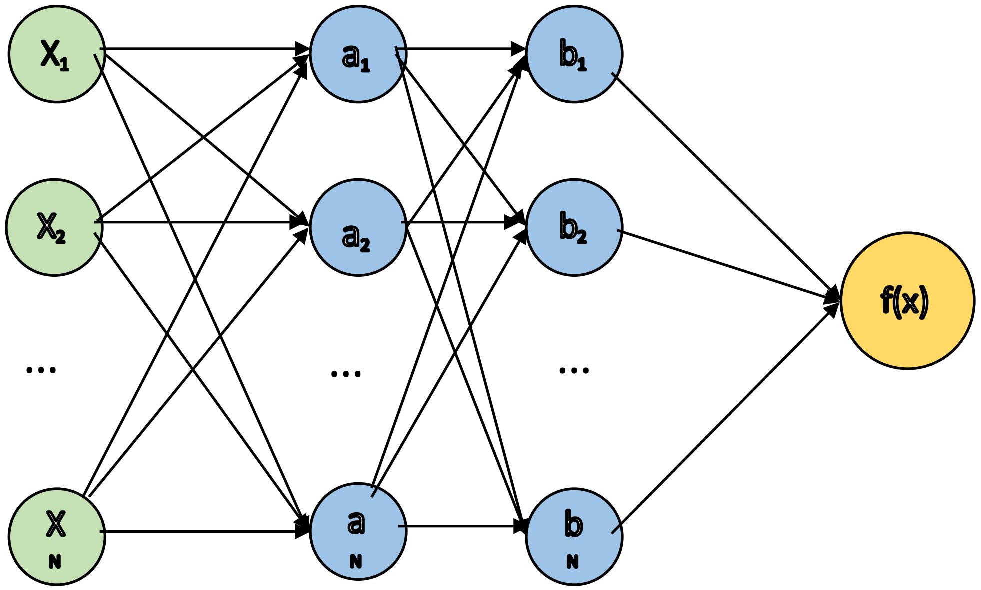
Source from: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

Machine Learning Models- Boosted D. Tree



- Each tree tries to minimize error from the previous tree in a sequential manner
- Final decision: Dtree1(x) + Dtree1(x) + Dtree1(x) + … + DtreeN(x)

Machine Learning Models - MLP

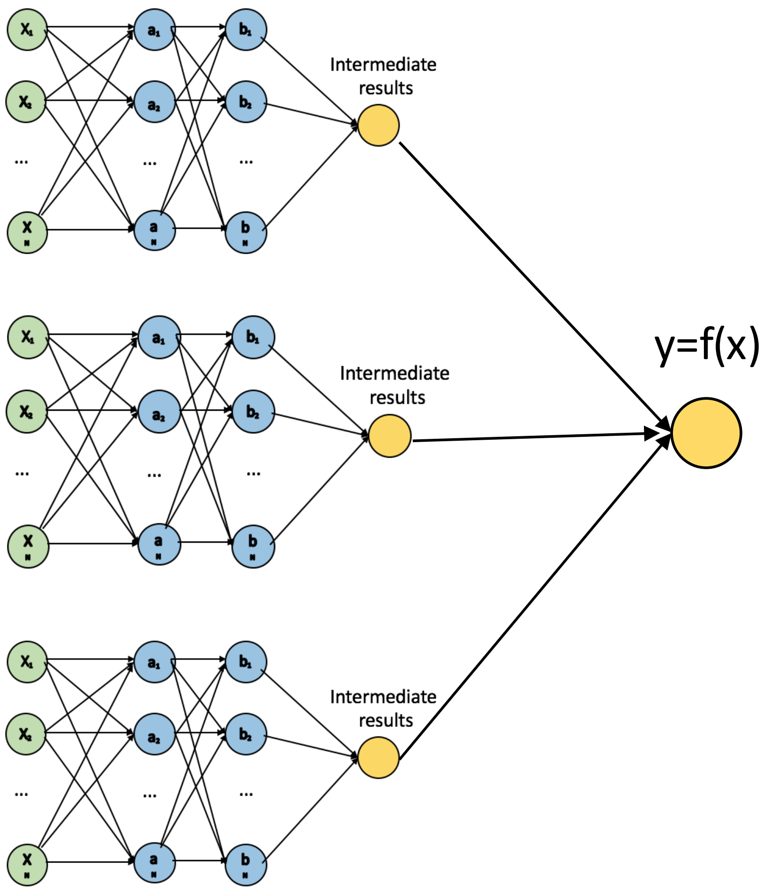


X_i: Input layers

a_i, b_i: Hidden layers
Each neuron computes:
$w_1a_1 + w_2a_2 + \ldots + w_Na_N$
w_1a_1 followed by non-linear activation function

y=f(x): Output layers

Machine Learning Models – MLP ensemble



Final Prediction
can be maximum,
minimum, median or
average

Classification using ML Algorithms

Classification Accuracy on Basic 6 Formats

		5 features - O(1)				*11 features - Sedaghati et al.*			
Machine	*precision*	Decs. Tree	SVM	MLP	XGBST	Decs. Tree	SVM	MLP	XGBST
K80c	*single*	60%	62%	62%	**67%**	81%	83%	83%	**85%**
	double	64%	63%	64%	**68%**	81%	85%	85%	**88%**
P100	*single*	65%	65%	67%	**69%**	79%	83%	82%	**84%**
	double	63%	65%	67%	**69%**	81%	83%	84%	**86%**

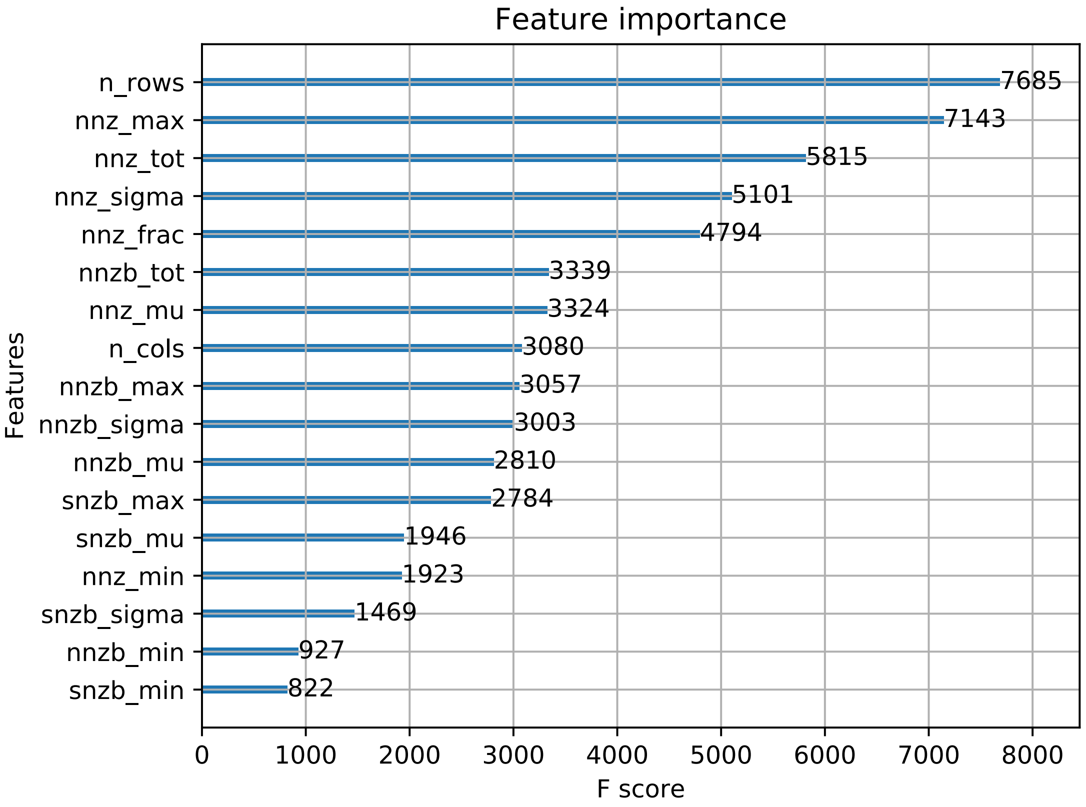
Classification accuracy on basic 6 formats: COO, ELL, CSR, HYB, CSR5 and merged-based CSR using feature sets 1 and 2 consisting of 11 features used in Sedaghati et el.

Classification Accuracy on Basic 6 Formats

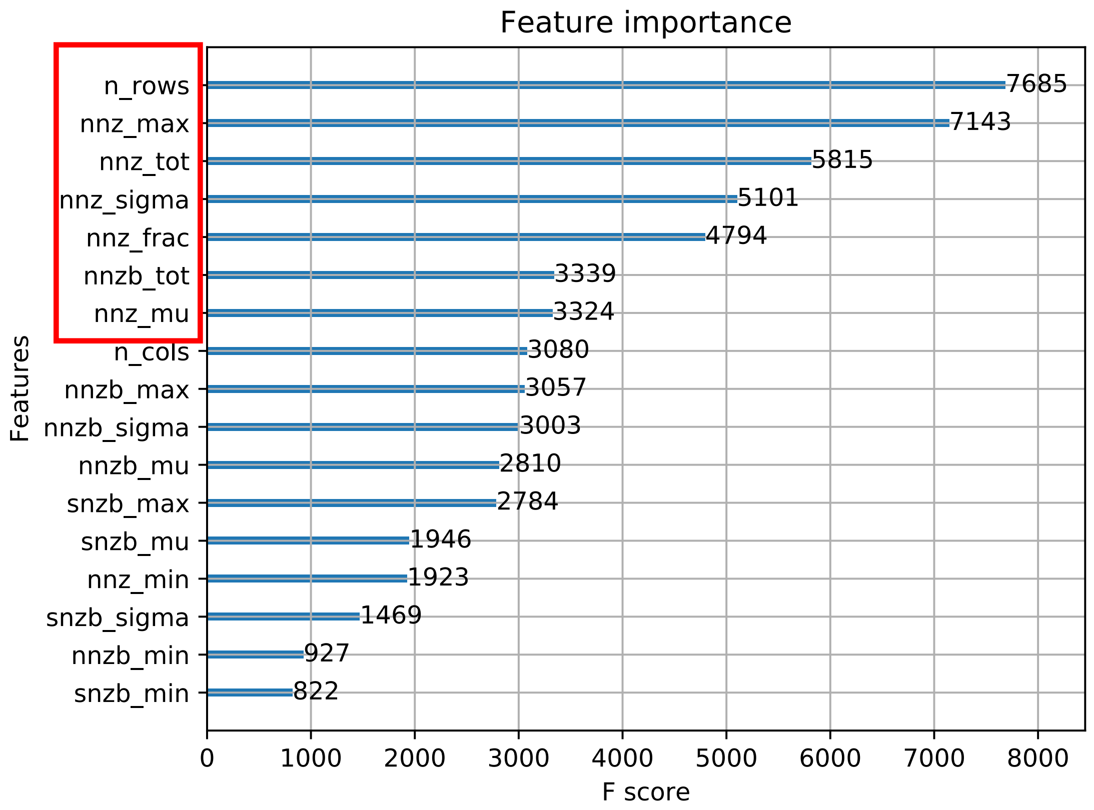
		11 features - Sedaghati et al.				*17 features*			
Machine	*precision*	Decs. Tree	SVM	MLP	XGBST	Decs. Tree	SVM	MLP	XGBST
K80c	*single*	81%	83%	83%	**85%**	78%	83%	83%	**85%**
	double	81%	85%	85%	**88%**	82%	85%	85%	**88%**
P100	*single*	79%	83%	82%	**84%**	79%	83%	82%	**84%**
	double	81%	83%	84%	**86%**	79%	83%	83%	**85%**

Classifcation accuracy on basic 6 formats: COO, ELL, CSR, HYB, CSR5 and merged-based CSR using feature sets 2 and 3

Feature Importance

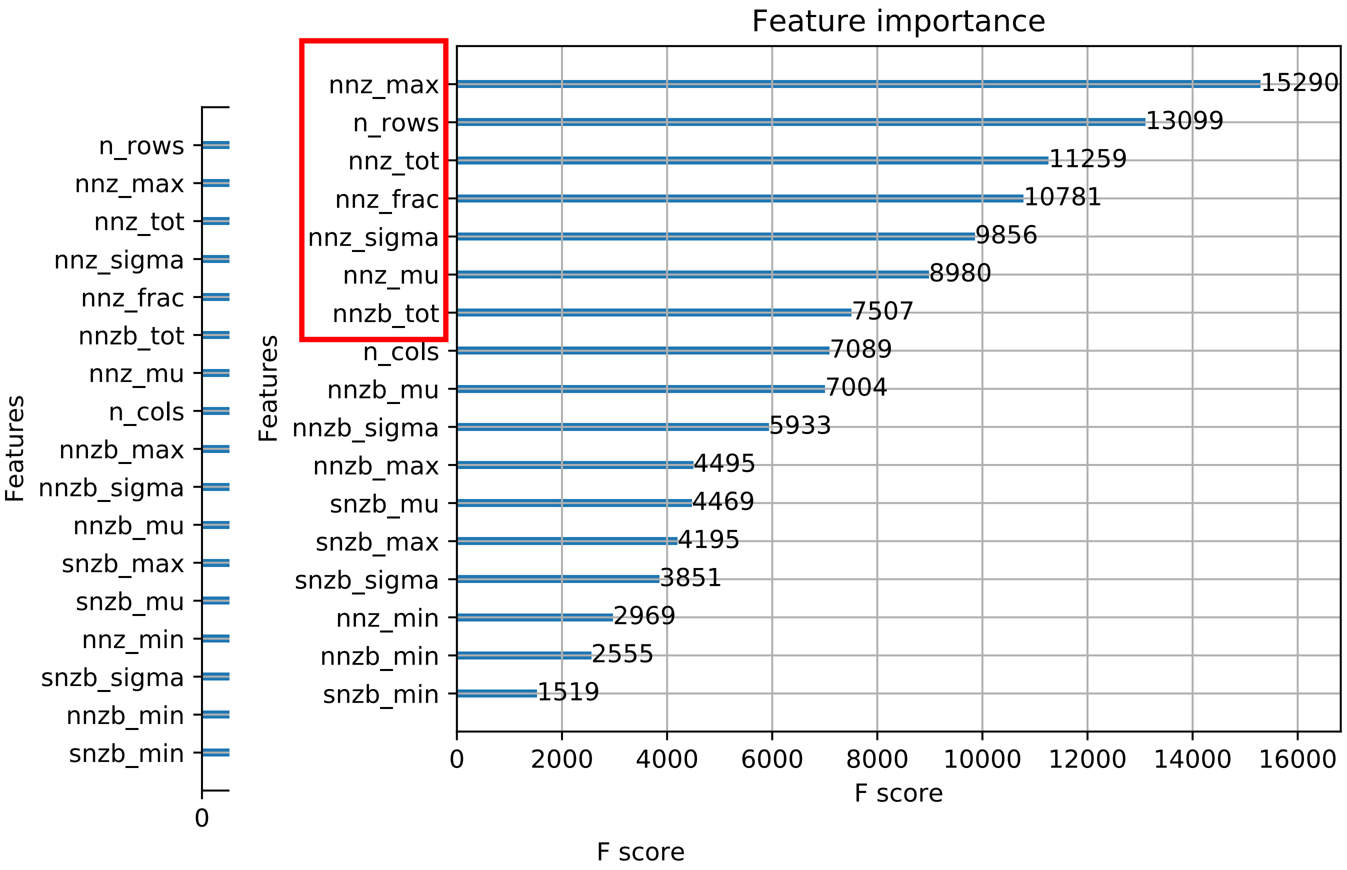


Feature Importance



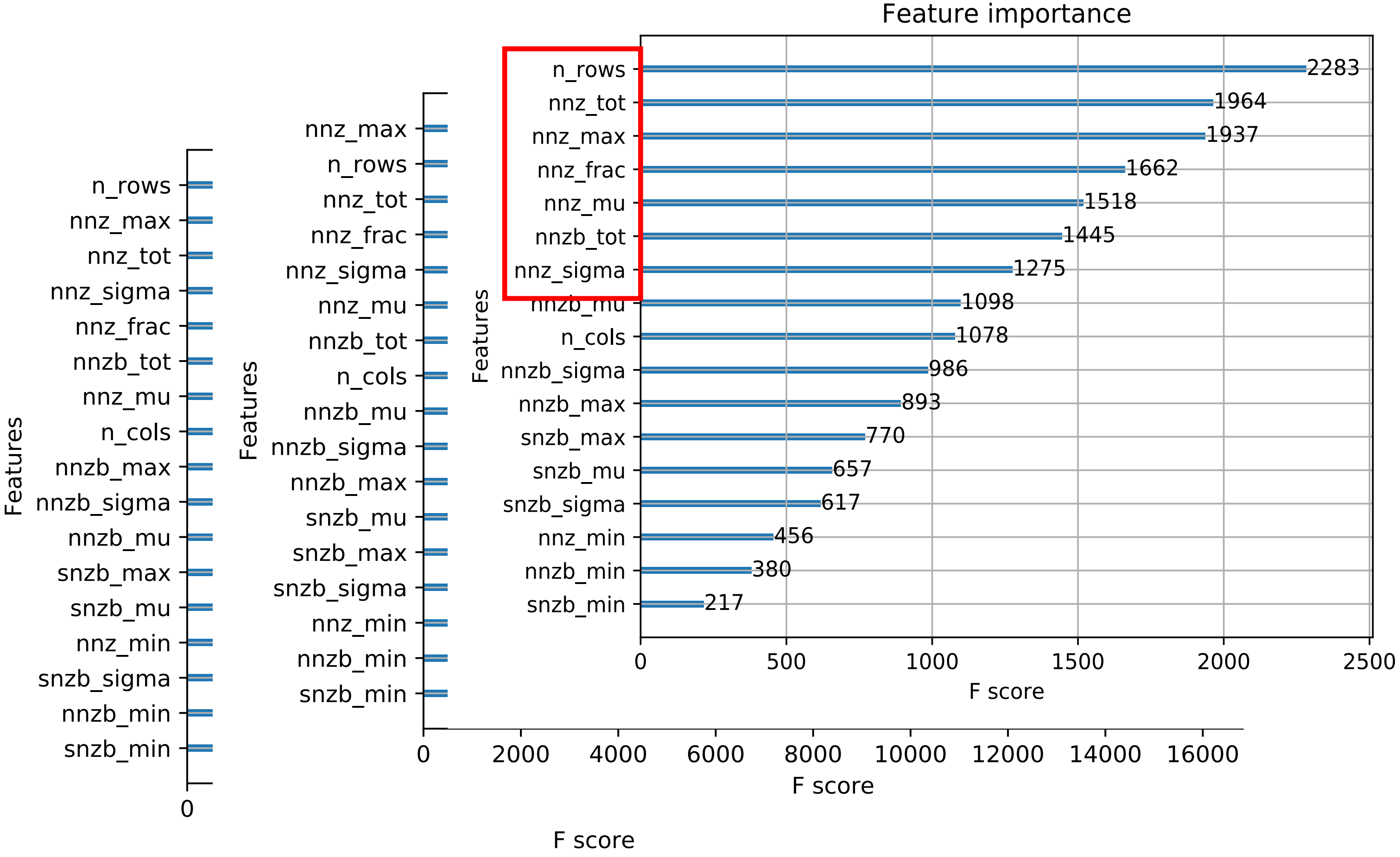
K80c - single precision

Feature Importance



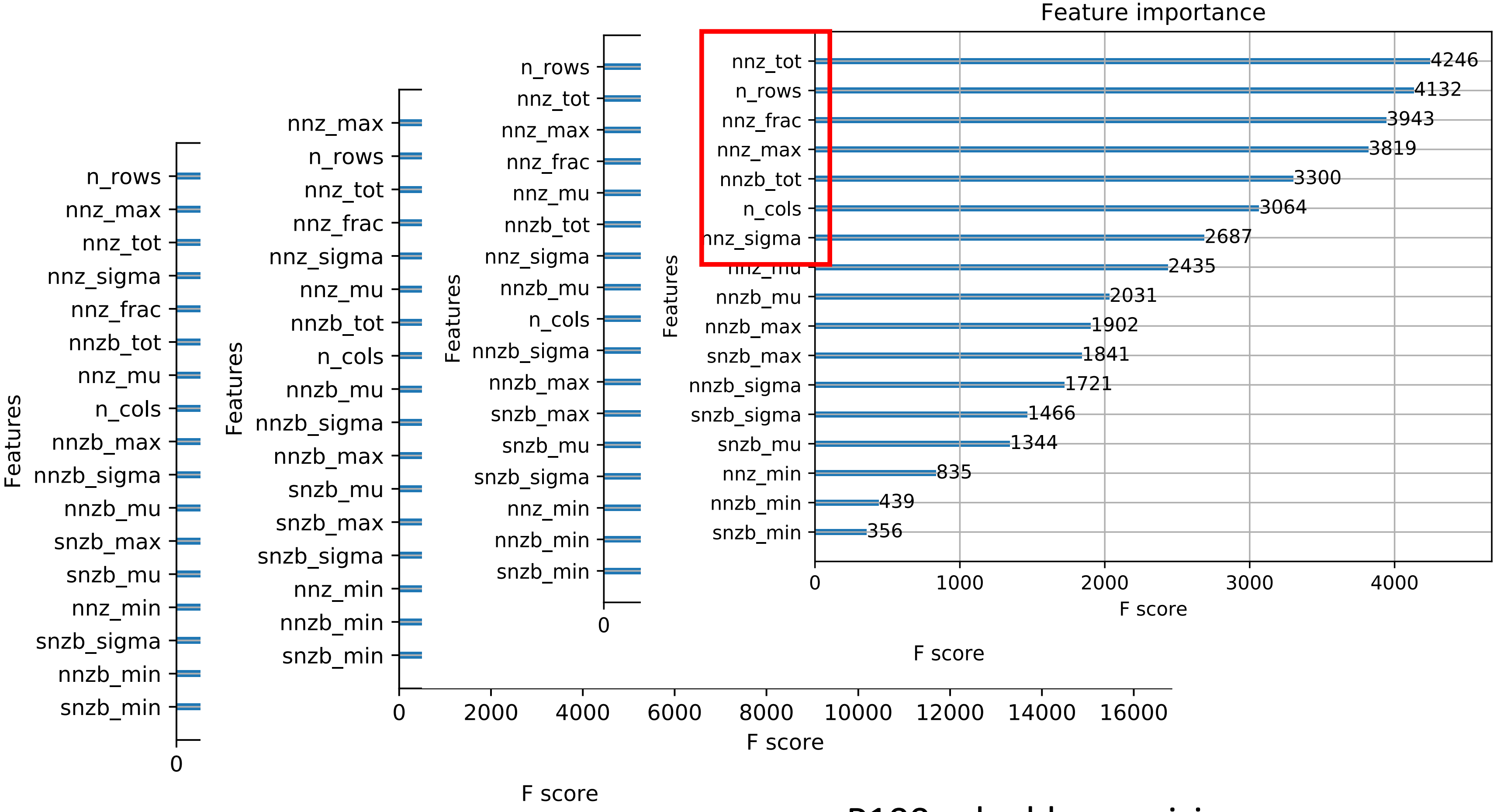
K80c - double precision

Feature Importance



P100 - single precision

Feature Importance



P100 - double precision

Classification using Top 7 features

		17 features				*Imp. (Top 7) features*			
Machine	*precision*	Decs. Tree	SVM	MLP	XGBST	Decs. Tree	SVM	MLP	XGBST
K80c	*single*	78%	83%	83%	**85%**	79%	85%	83%	**85%**
	double	82%	85%	85%	**88%**	83%	87%	86%	**88%**
P100	*single*	79%	83%	82%	**84%**	77%	83%	83%	**84%**
	double	79%	83%	83%	**85%**	79%	84%	85%	**86%**

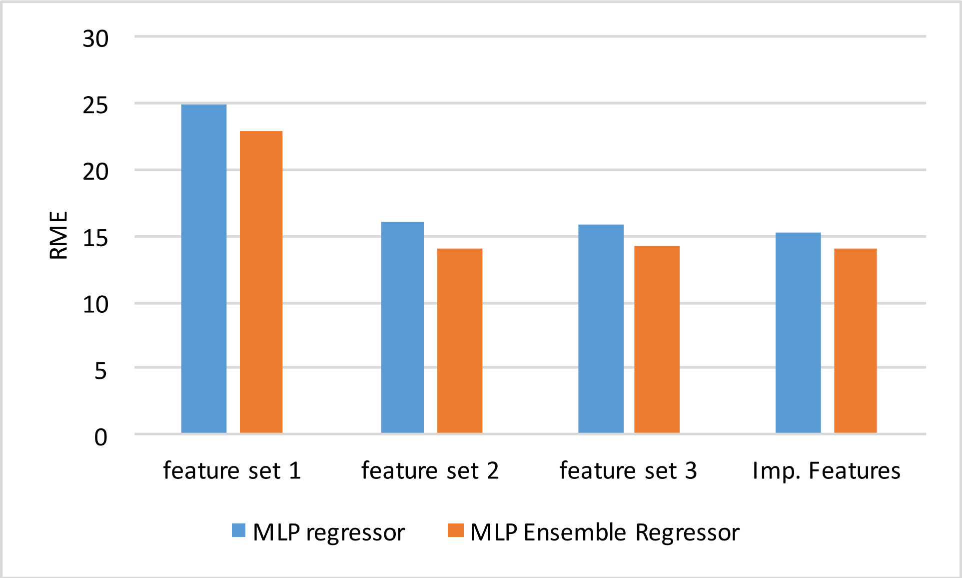
Classication accuracy on basic 6 formats: COO, ELL, CSR, HYB, CSR5 and merged-based CSR using feature sets 2 and Imp. features

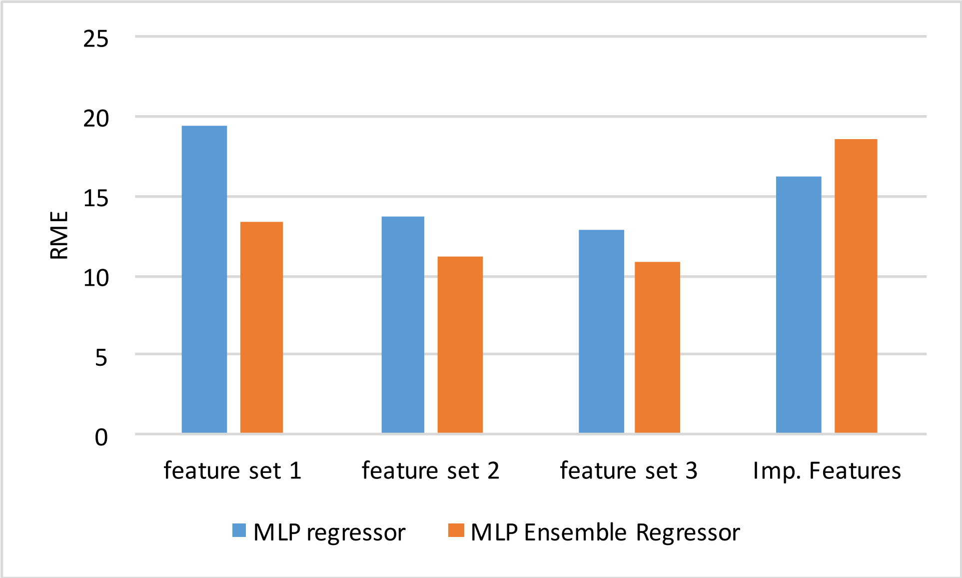
Performance Modeling of SpMV using ML Algorithms

Performance Modeling

- Conventional methods are based on analytical modeling
- GPU's complicated architecture
- Detailed knowledge of the architecture required
- Can simple ML algorithms also predict performance of various SpMV formats?

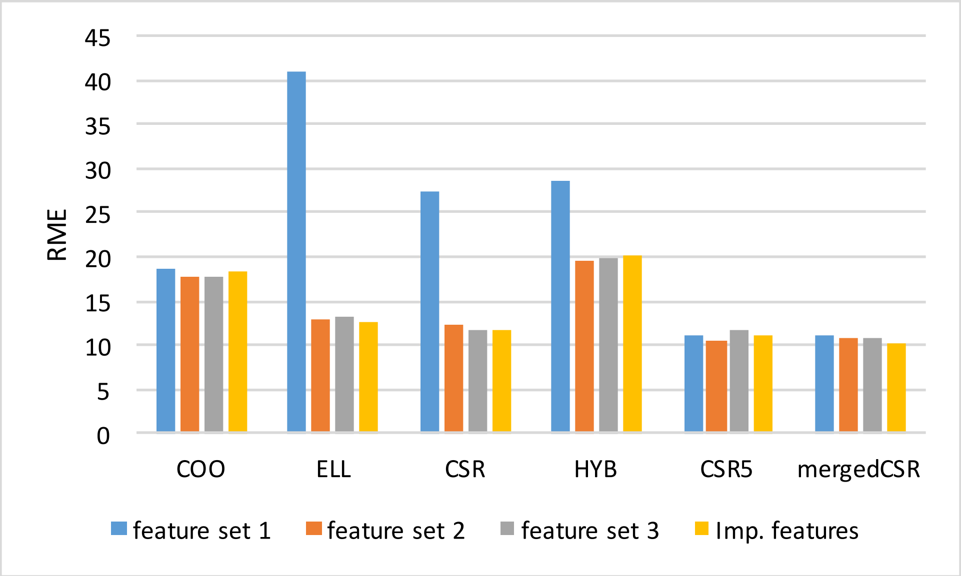
Average Relative Mean Error (RME)

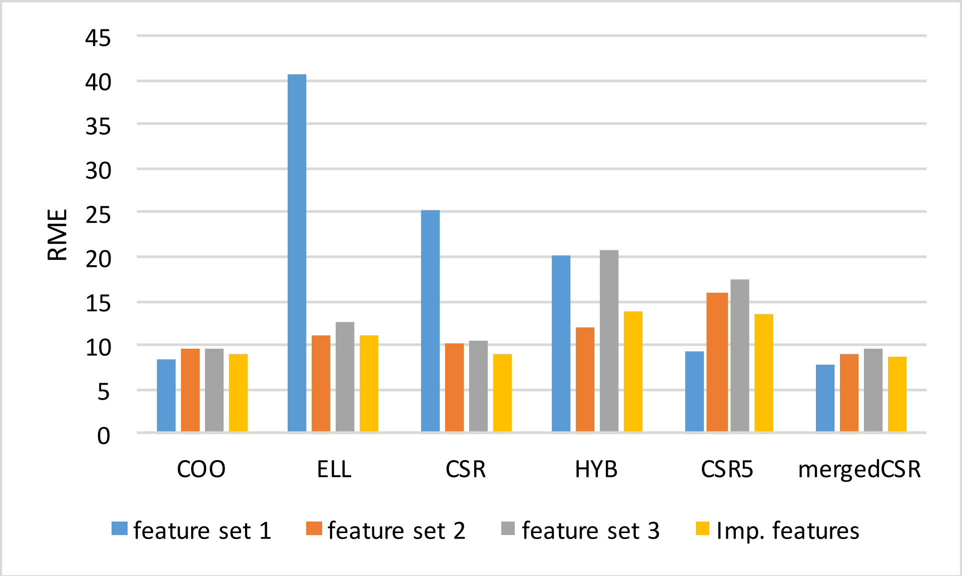




Average relative mean error (RME) of 6 formats using MLP and ML ensemble regressor on Tesla K80c and Tesla P100 GPU using double precision data type

RME for Each Format





Relative mean error (RME) achieved by each 6 formats using MLP ensemble regressor on Tesla K80c and Tesla P100 GPU using double precision data type

Conclusion

- XGBoost achieves the highest classification accuracy
- List of 7 features which are sufficient to provide the best classification accuracy
- MLP-ens, a simple neural network model to predict the performance of a given input matrix

Thank you!