Using Machine Learning to Simplify the Identification of Code Optimization

Rohith Kumar Uppala
SIParCS Intern 2018

Mentors:
Youngsung Kim and **John Dennis**

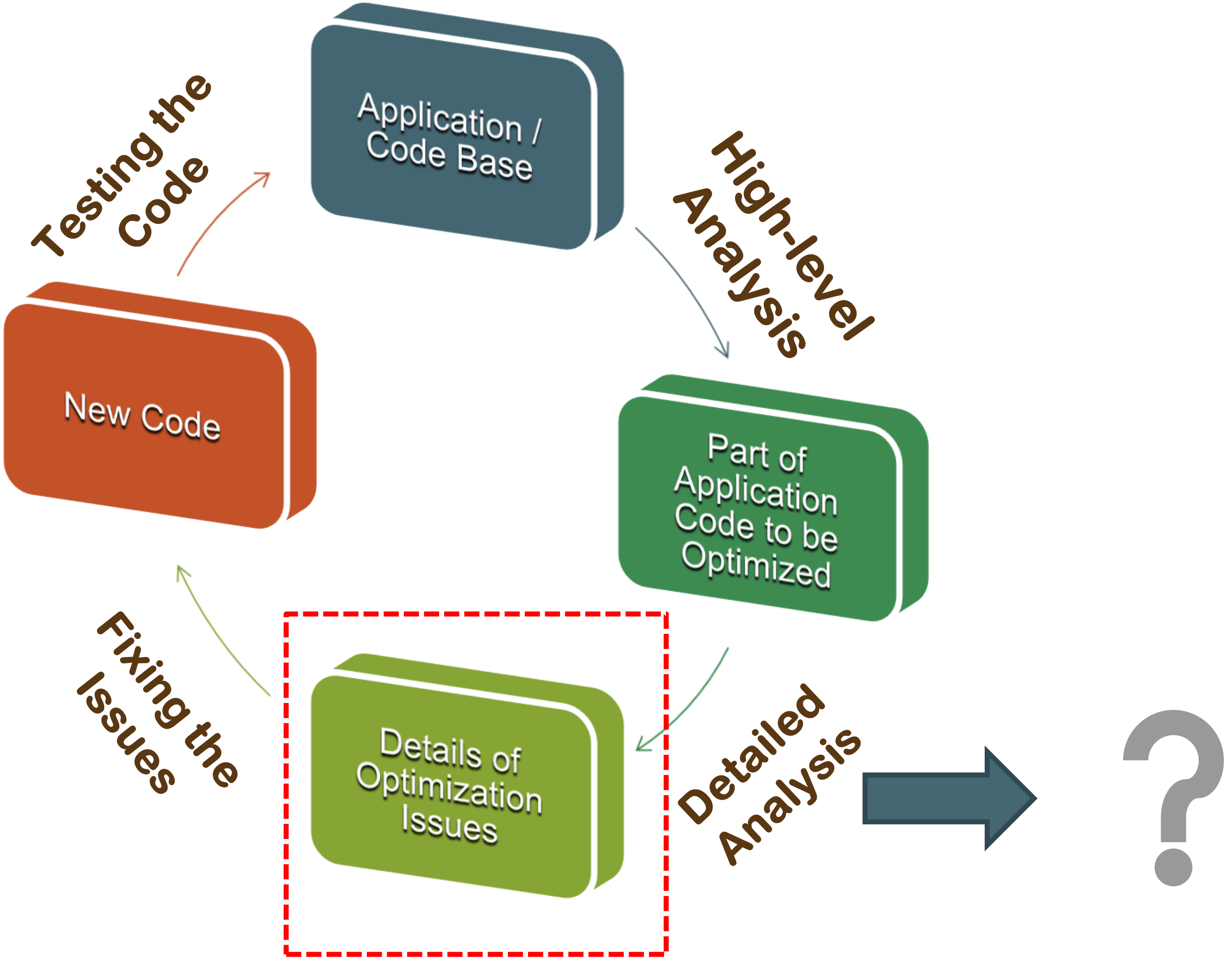
NCAR
August 3, 2018

Code Optimization

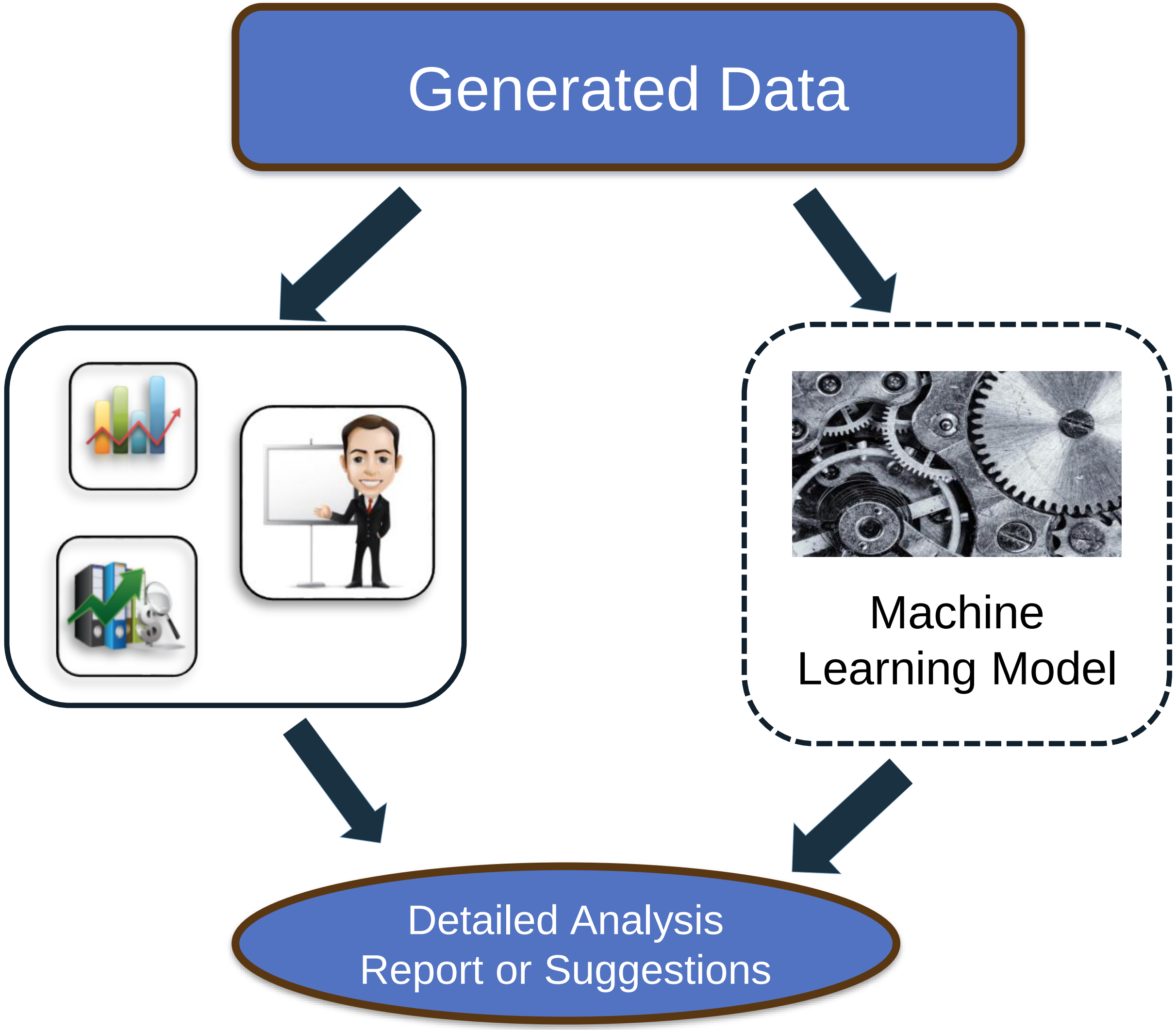
- What is code Optimization ?
  - Code optimization is any method of code modification to improve performance and efficiency
  - It can refer to
    - Optimizing the code for efficiency
    - Reducing the lines of code for readability

- Why ?
  - Smaller size
  - Consume less memory
  - Execute more rapidly
  - Perform fewer input/output operations
  - On shared resources, end to end job throughput may increase super linearly with speedup

Optimization is an Iterative Process



Motivation

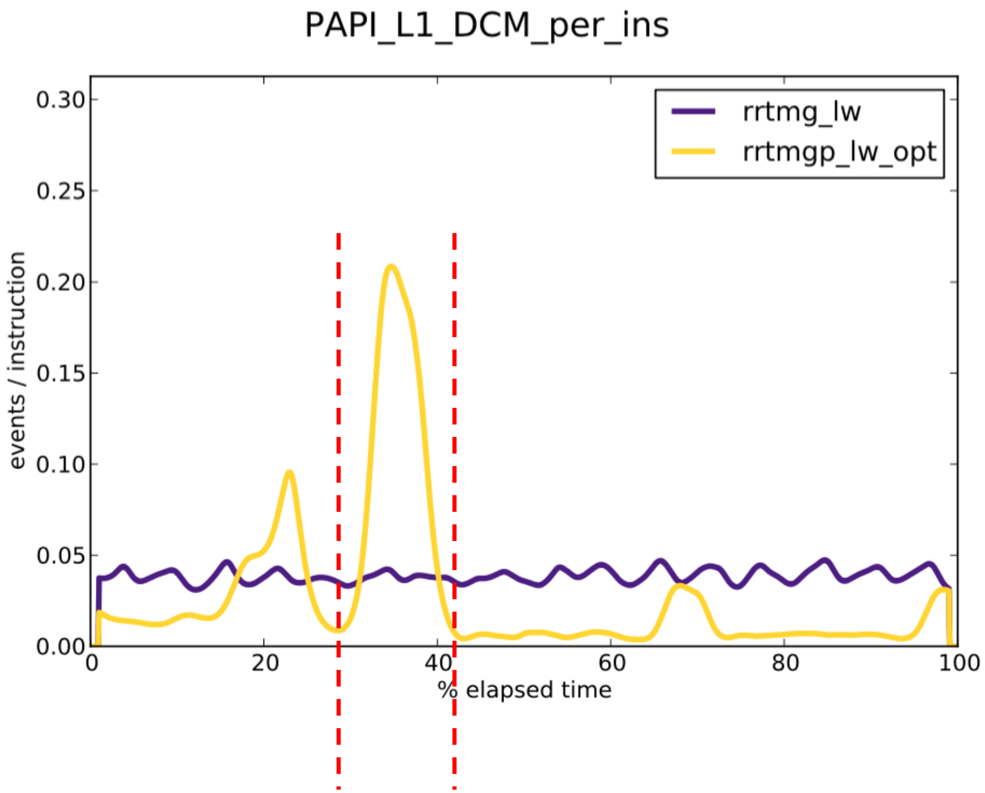


Generated Data

Machine Learning Model

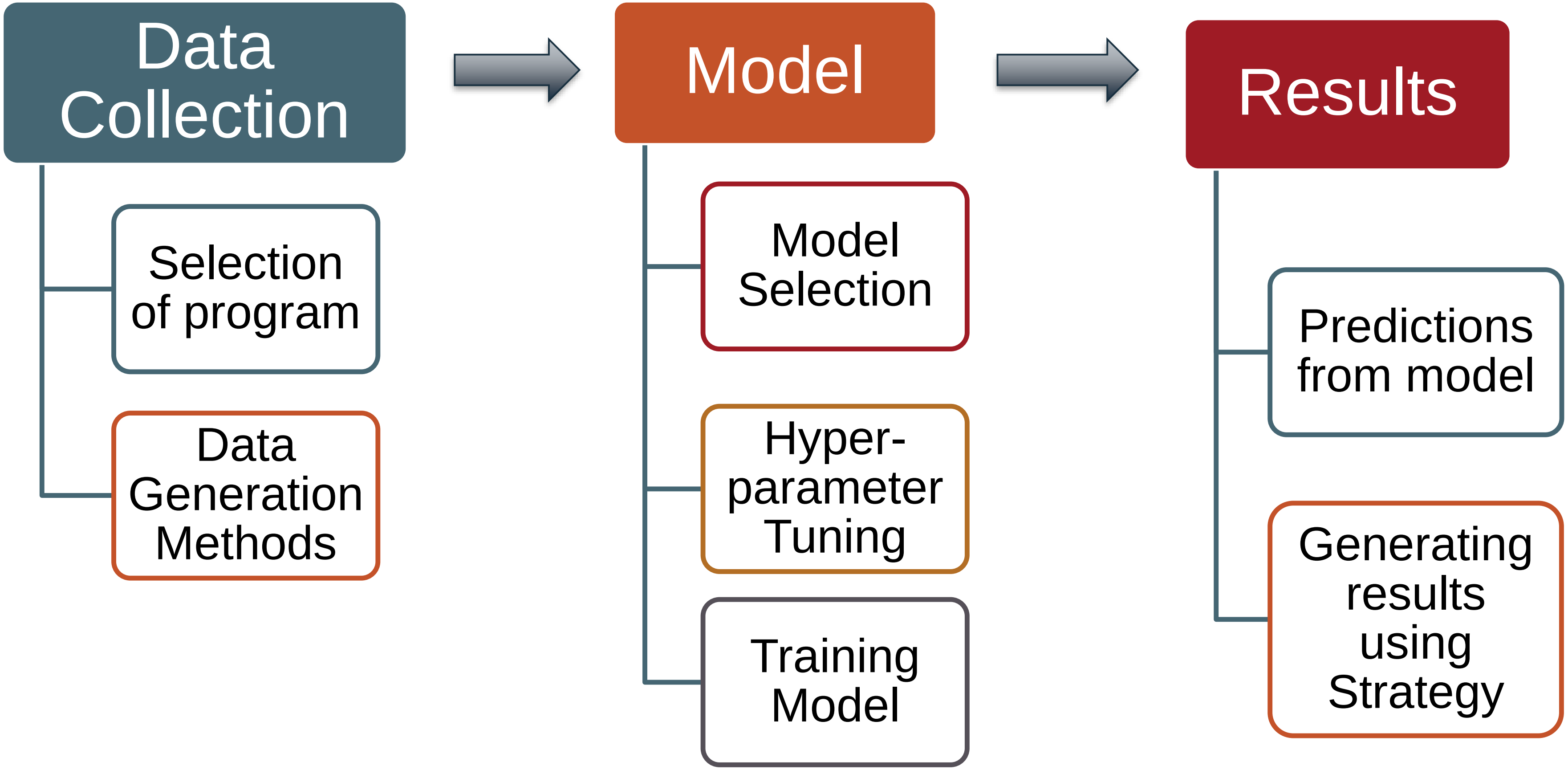
Detailed Analysis Report or Suggestions

Example



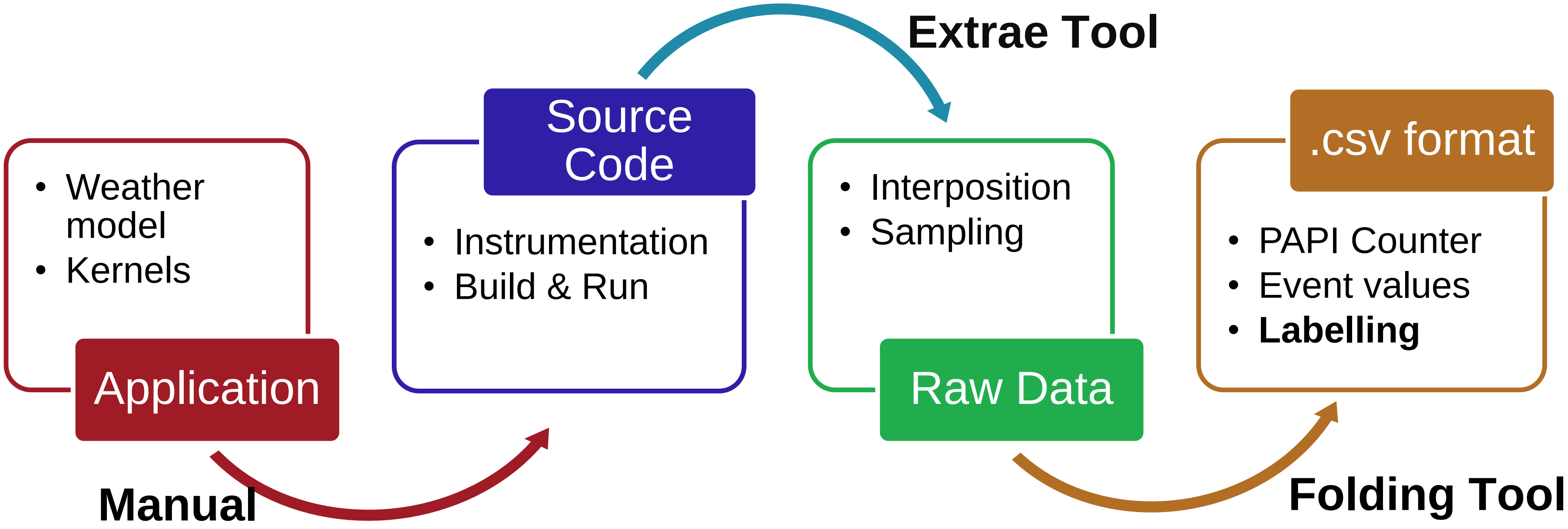
- Select the region based on events per instruction
- Map the samples in the region with Line ID and time ID
- Get the Line Number and File Name from Line ID

Project Overview



- Data Collection
  - Selection of program
  - Data Generation Methods
- Model
  - Model Selection
  - Hyper-parameter Tuning
  - Training Model
- Results
  - Predictions from model
  - Generating results using Strategy

Collecting the Data



Application

- Weather model
- Kernels

Manual

Source Code

- Instrumentation
- Build & Run

Extrae Tool

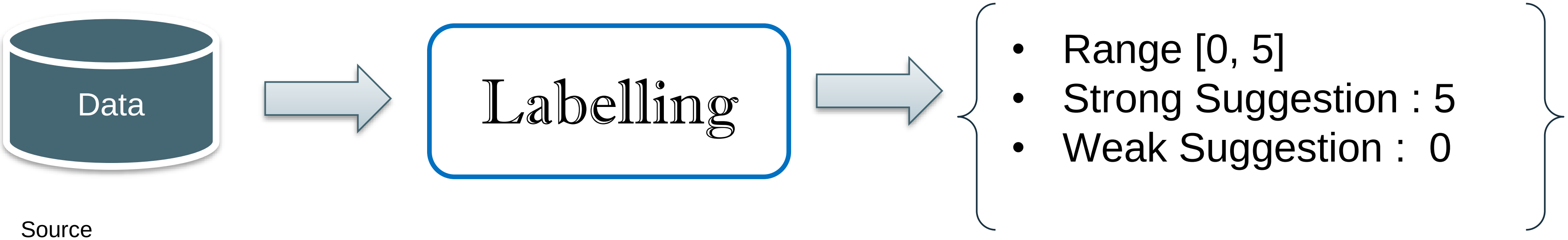
Raw Data

- Interposition
- Sampling

Folding Tool

.csv format

- PAPI Counter
- Event values
- **Labelling**



Data

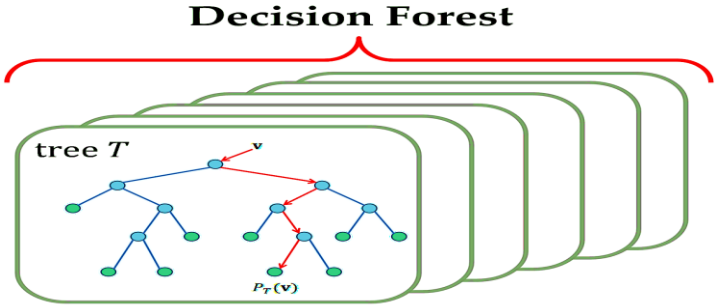
Labelling

- Range [0, 5]
- Strong Suggestion : 5
- Weak Suggestion : 0

Source
Extrae Tool : https://tools.bsc.es/extrae
Folding Tool : https://tools.bsc.es/folding

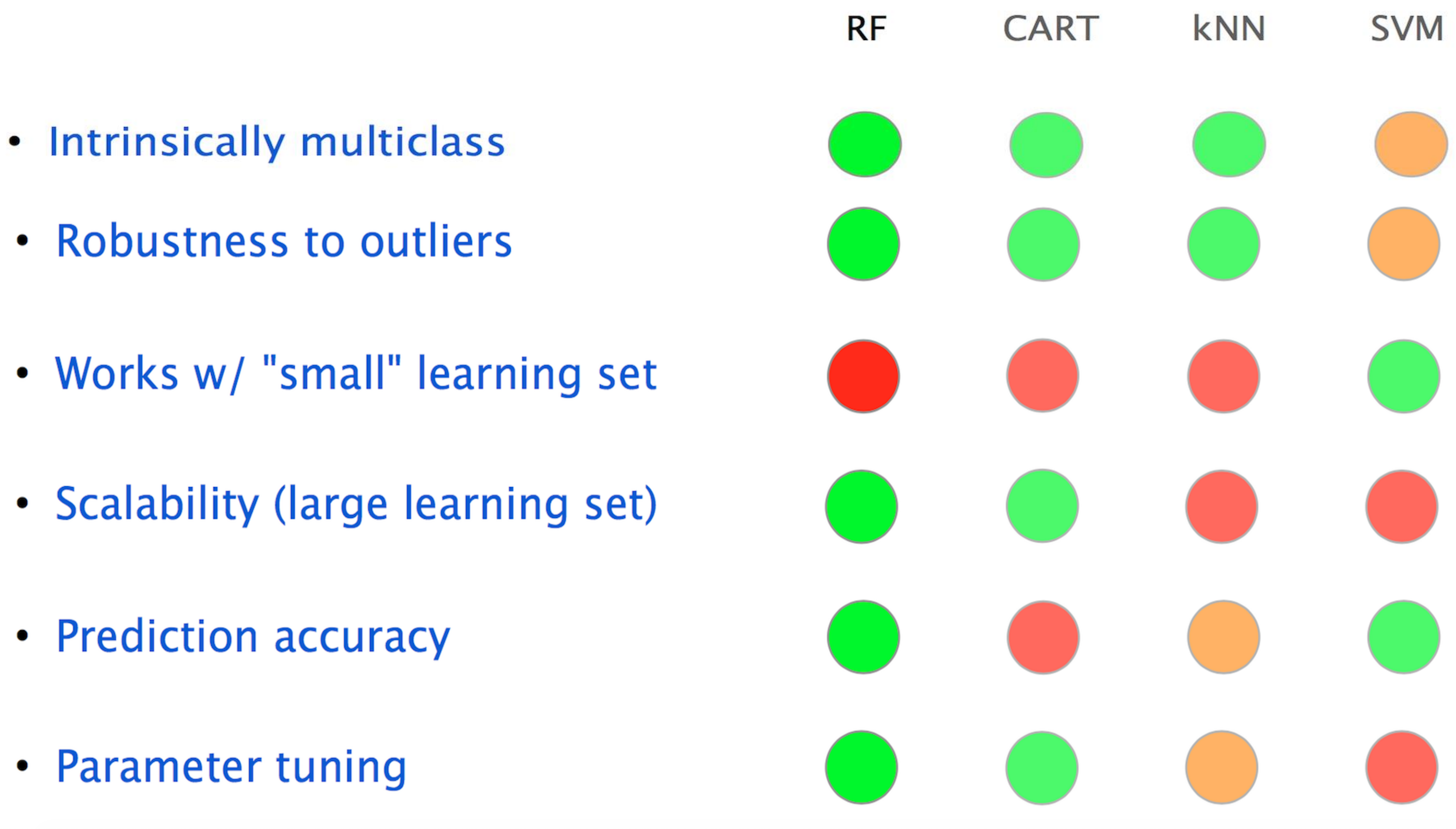
Selecting the Model

- This is a Supervised Classification and Regression task.
  - Random Forest
  - Classification and Regression Tree
  - Support Vector Machine
  - K-Nearest neighbors



- Advantages of Random Forest over other models
  - Can handle categorical features very well
  - Less prone to overfitting
  - It can handle high dimensional spaces as well as large number of training examples
  - It works for almost any type of classification tasks

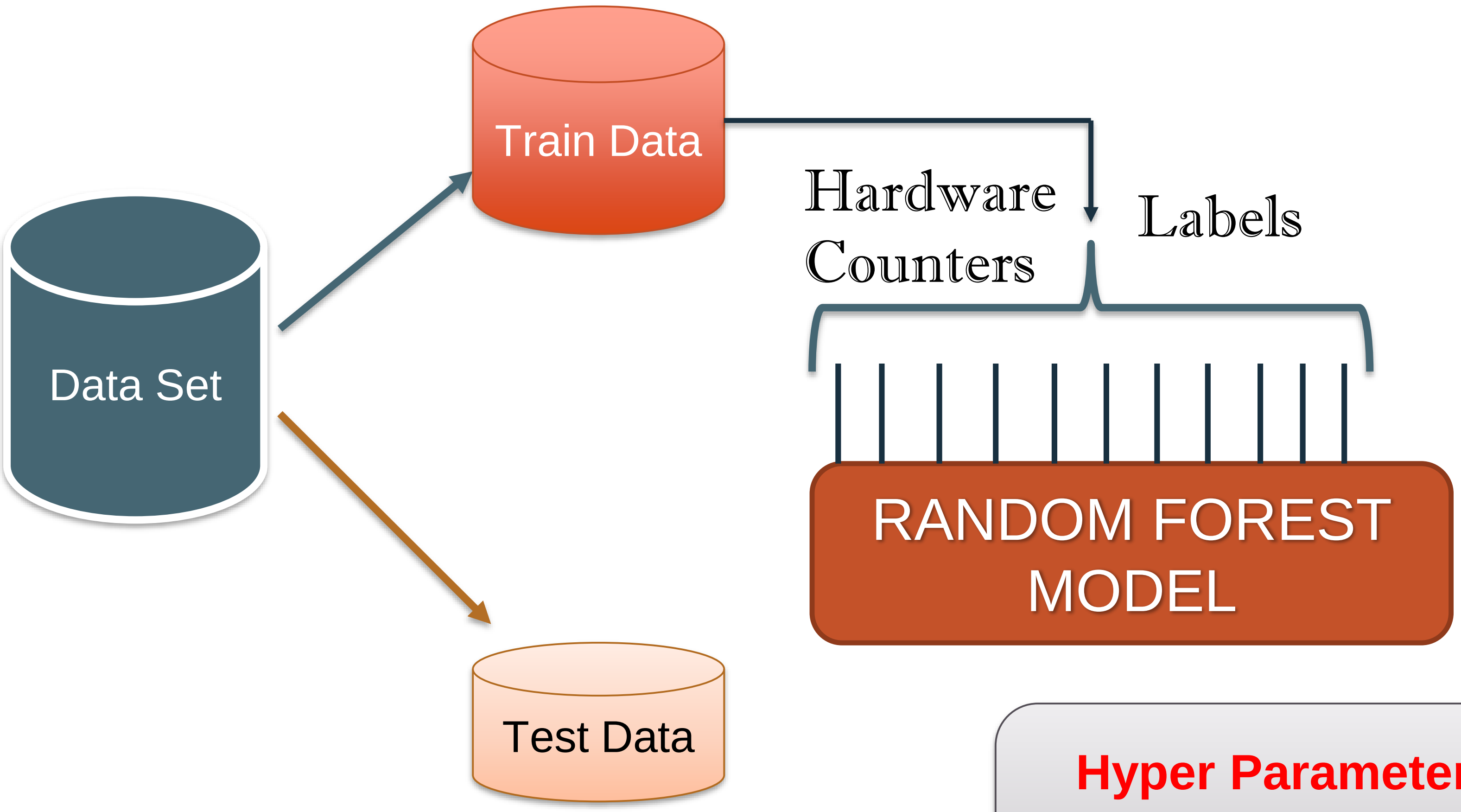
Model Comparisons



	RF	CART	kNN	SVM
Intrinsically multiclass	🟢	🟢	🟢	🟠
Robustness to outliers	🟢	🟢	🟢	🟠
Works w/ "small" learning set	🔴	🔴	🔴	🟢
Scalability (large learning set)	🟢	🟢	🔴	🔴
Prediction accuracy	🟢	🔴	🟠	🟢
Parameter tuning	🟢	🟢	🟠	🔴

Source: An Introduction to random forests by Eric Debreuve/ Team Morpheme

Training the Models

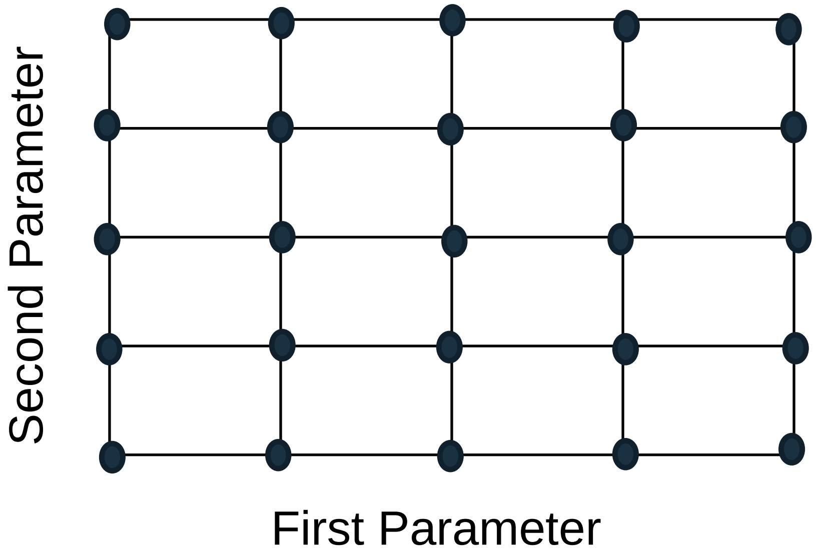


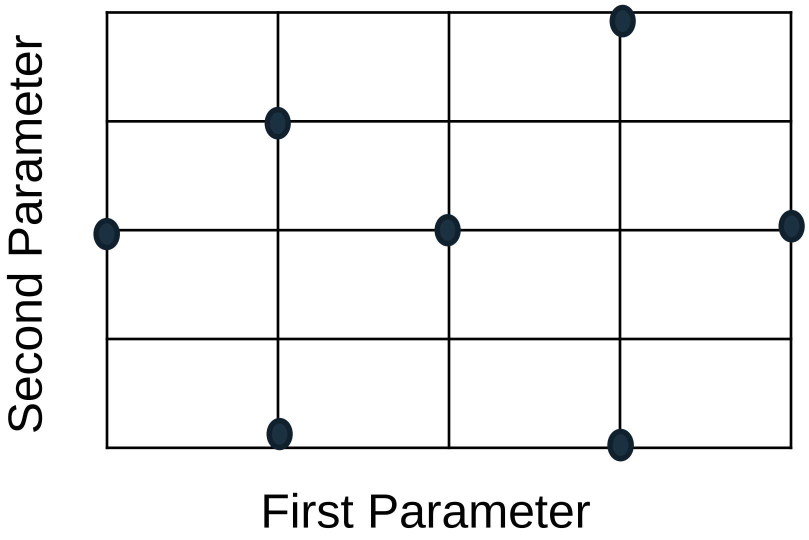
Hyper Parameters

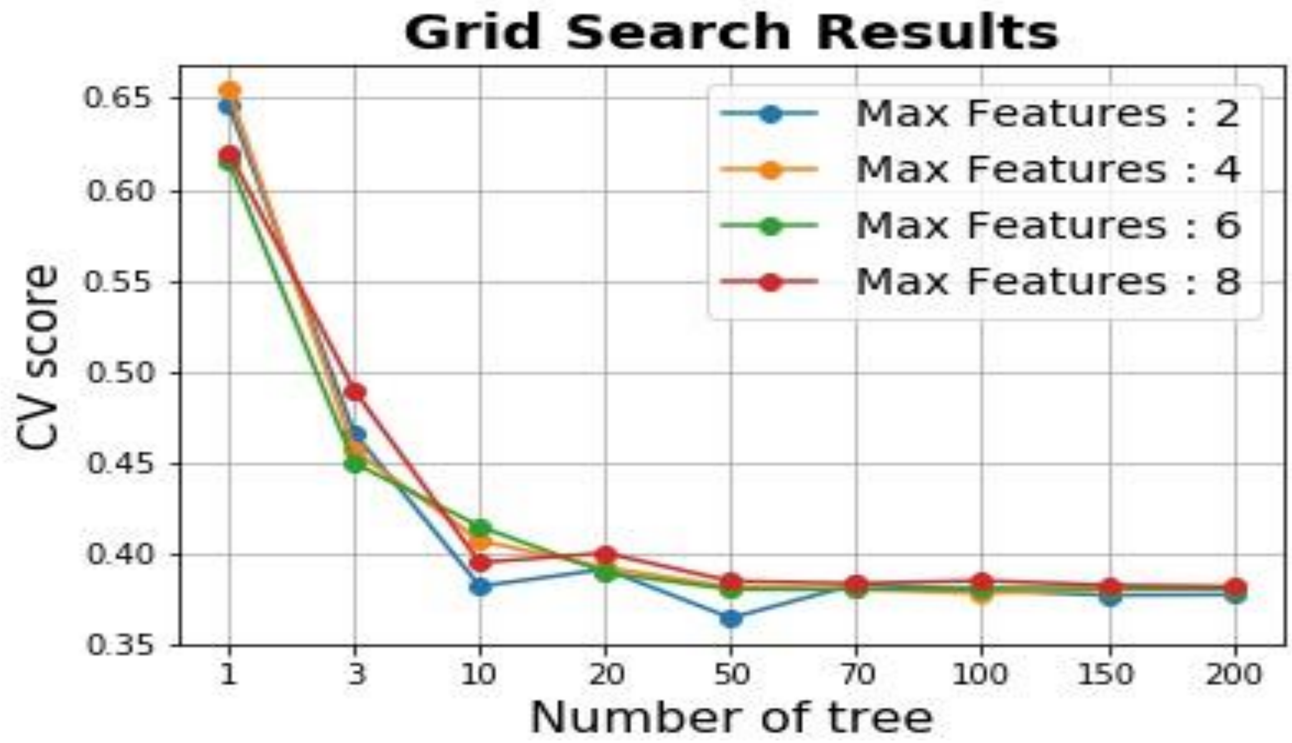
- n_trees = ?
- max_features = ?

Hyper-Parameters

- **Traditional Approach :** manual tuning
  - With expertise in machine learning algorithms and their parameters, the best settings are directly dependent on the data used in the training and scoring
- **Hyperparameter Optimization :** grid vs random



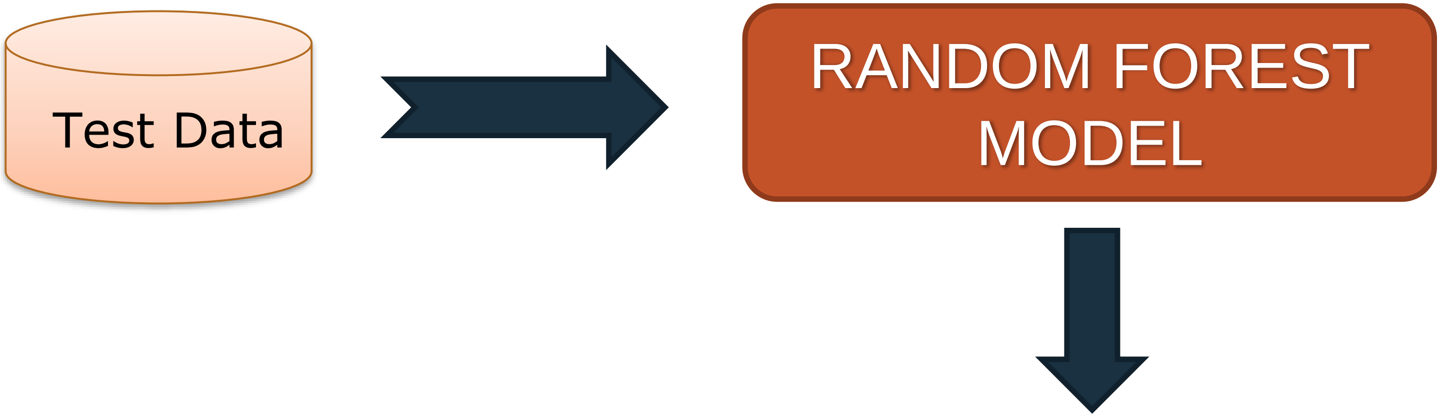




- We selected Grid Search Cross Validation because we are dealing with relatively small dataset size
- Parameters with the lowest Cross Validation score are best Parameters

Final Parameters : Max Features. : 2 and Number of Trees. : 50

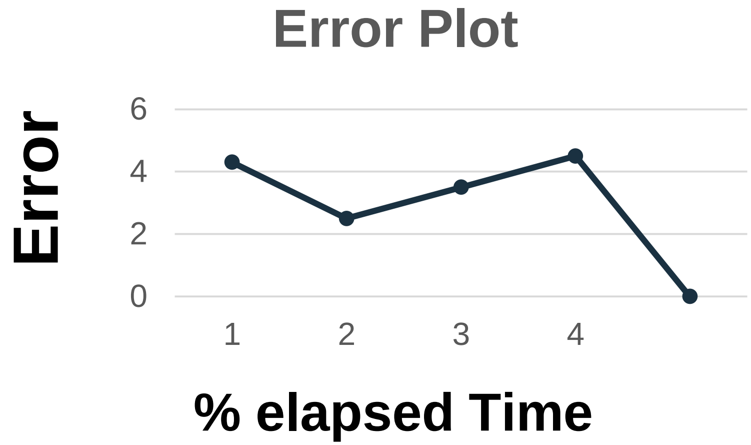
Testing the Models



Test Data

RANDOM FOREST MODEL

Predicted Labels

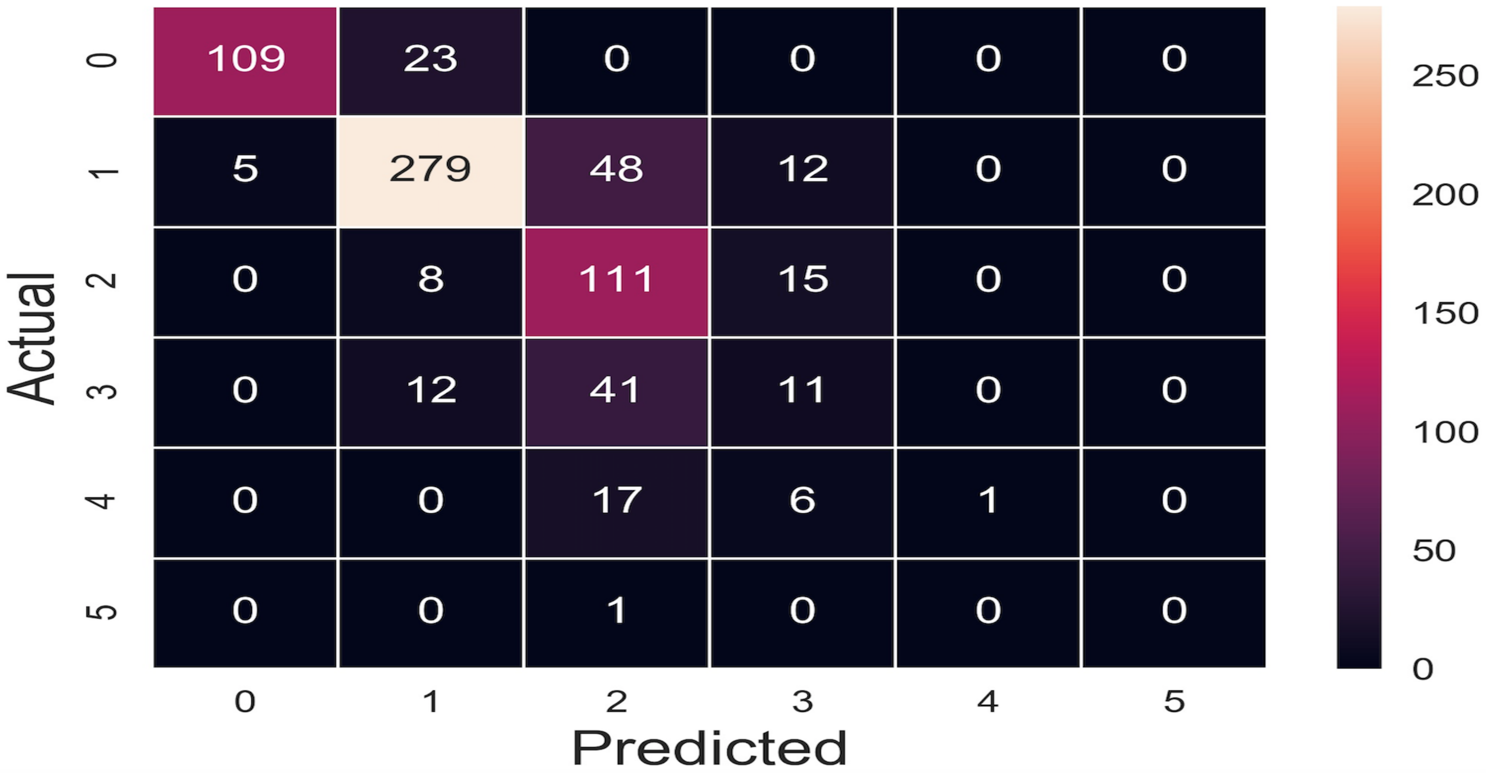


Error =

Actual – Predicted

Confusion Matrix

A **Confusion Matrix** is a table used to described the performance of a classification model on a set of test data for which the true values are known



Actual \ Predicted	0	1	2	3	4	5
0	109	23	0	0	0	0
1	5	279	48	12	0	0
2	0	8	111	15	0	0
3	0	12	41	11	0	0
4	0	0	17	6	1	0
5	0	0	1	0	0	0

Precision and Recall

Actual	True Negative	False Positive
	False Negative	True Positive
	Predicted	

- Multiple statistics are often computed from a confusion matrix for a binary classifier

$$\text{Precision} = \frac{True\ Positive}{True\ Positive+False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive+False\ Negative}$$

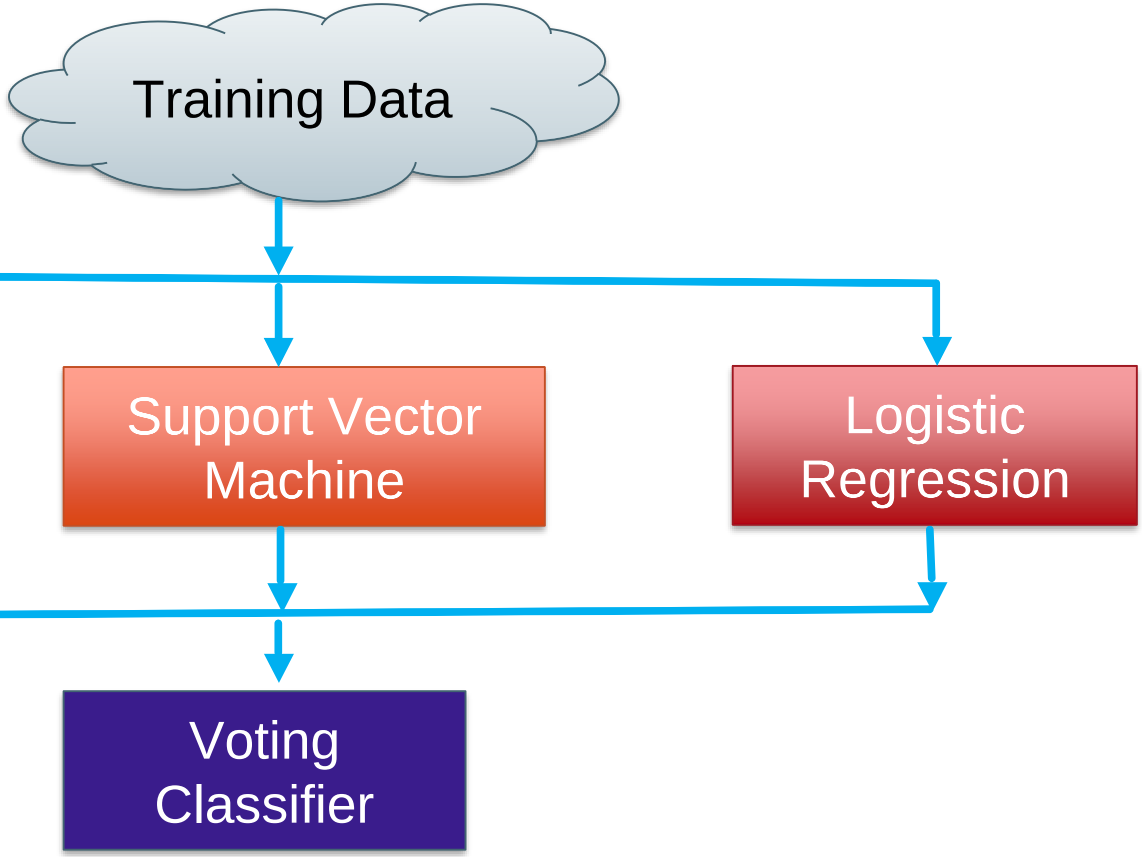
$$\text{RMSE} = \surd\left(\frac{\sum_0^n (y' - y)}{n}\right)$$

Results for Test Set

- Root mean Square Error : 0.59
- Precision : 0.803
- Recall : 0.770

Wisdom Of the Crowd

- Aggregated results > best single classifier result
- Basic idea is to learn a set of classifiers and to allow them to vote



Comparison of Classifiers

Random Forest Classifier

Actual \ Predicted	0	1	2
0	98	464	0
1	42	288	0
2	0	8	0

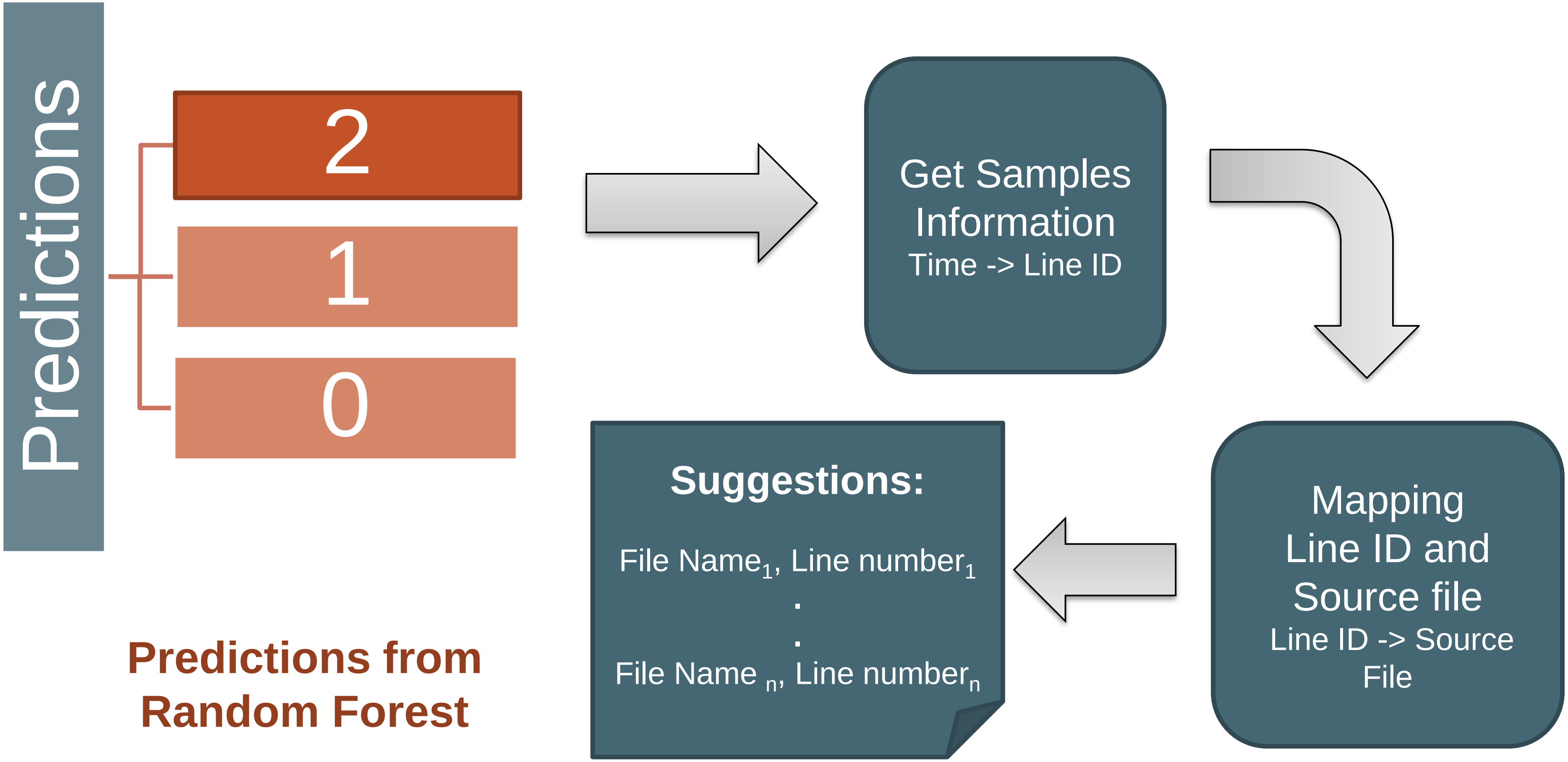
- Precision : 0.511
- Recall : 0.498

Voting Classifier

Actual \ Predicted	0	1	2
0	100	435	27
1	6	296	28
2	0	0	8

- Precision : 0.726
- Recall : 0.47

Generating Suggestions



Predictions

2

1

0

Predictions from Random Forest

Get Samples Information

Time -> Line ID

Mapping Line ID and Source file

Line ID -> Source File

Suggestions:

File Name_1, Line number_1

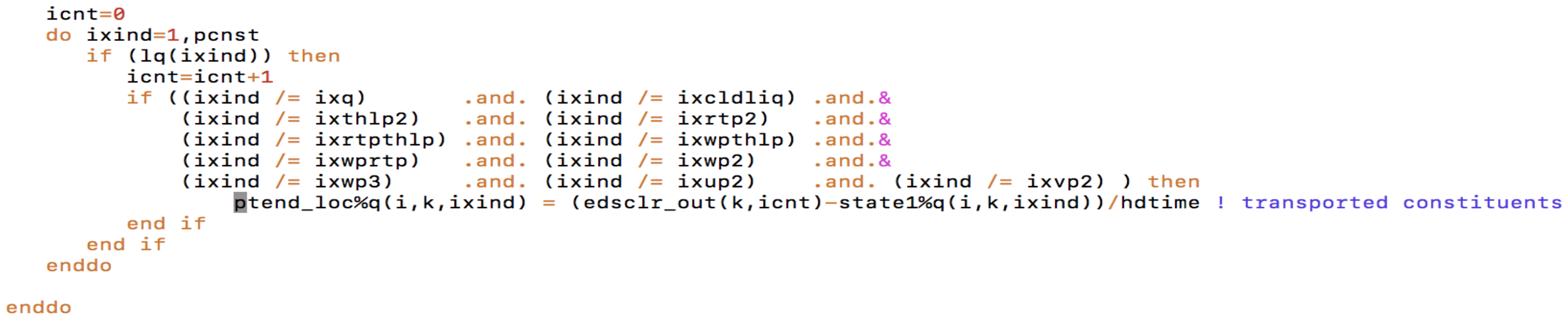
.

.

File Name $_n$, Line number_n

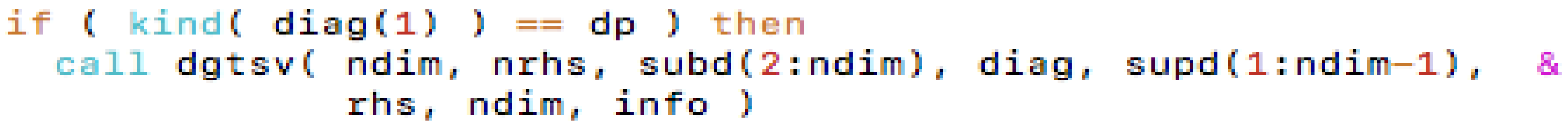
Results

- *clubb_intr.F90 , 2801*



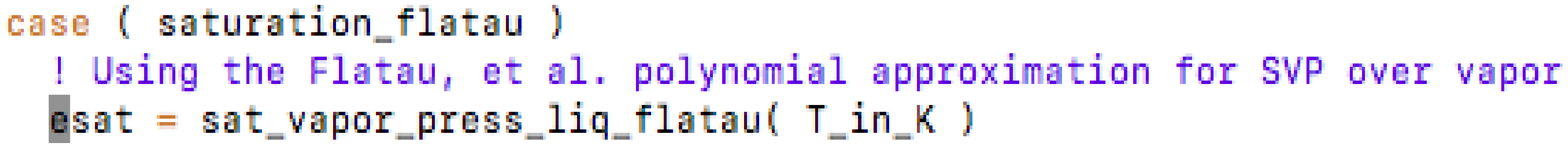
```
   icnt=0
   do ixind=1,pcnst
      if (lq(ixind)) then
         icnt=icnt+1
         if ((ixind /= ixq)       .and. (ixind /= ixcldliq) .and.&
             (ixind /= ixthlp2)   .and. (ixind /= ixrtp2)   .and.&
             (ixind /= ixrtpthlp) .and. (ixind /= ixwpthlp) .and.&
             (ixind /= ixwprtp)   .and. (ixind /= ixwp2)    .and.&
             (ixind /= ixwp3)     .and. (ixind /= ixup2)    .and. (ixind /= ixvp2) ) then
                 ptend_loc%q(i,k,ixind) = (edsclr_out(k,icnt)-state1%q(i,k,ixind))/hdtime ! transported constituents
         end if
      end if
   enddo

enddo
```

- *lapack_wrap.F90 265*



```
if ( kind( diag(1) ) == dp ) then
  call dgtsv( ndim, nrhs, subd(2:ndim), diag, supd(1:ndim-1),  &
              rhs, ndim, info )
```

- *saturation.F90 175*



```
case ( saturation_flatau )
  ! Using the Flatau, et al. polynomial approximation for SVP over vapor
  esat = sat_vapor_press_liq_flatau( T_in_K )
```

Future Work

- Currently we are generating suggestions based only on the vectorization method, we want to add other optimization techniques

- Work with other datasets and get optimal results for error, precision and recall score

- We are curious to see results from how Dimensionality Reduction can affect our prediction and speed up the process

Acknowledgements

- Youngsung Kim, ~~Magicians~~, Mentor
- John Dennis, ~~Magicians~~, Mentor
- Brian Dobbins
- Rich Loft
- AJ Lauer
- Elliot Foust
- Elizabeth Faircloth
- Jenna Preston

- SIParCS
- CISL
- NSF
- NCAR

Special Thanks to :

- All fellow interns
- Shuttle Drivers

Thank You

Any Questions ?

Email : rohithuppala28@gmail.com