What is Gode clone: Two code fragments form a clone pair if they are similar enough according to a given definition of simularity

1/5=10h :

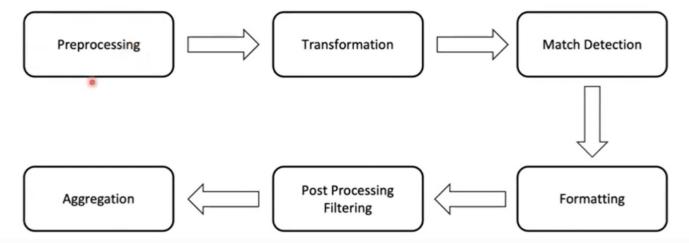
```
Type 2
                                                                                                            Type 3
Type 1
                                                         public int sum(int a, int b){
   int sum;
   sum = a + b;
   return sum;
}
   public int sum(int a, int b){
  int sum;
  sum = a + b;
                                                                                                                 public int sum(int a, int b){
                                                                                                                     int sum;
sum = a + b;
                                                                                                                     return sum;
                                                      public int sum(int num1, int num2){
  int result;
  result = num1 + num2;
  return result;
   public int sum(int a, int b){
                                                                                                                public int sum(int a, int b){
   return a + b;
        int sum;
sum = a + b;
return sum;
       Identical code fragments
                                                              Identical code fragments
                                                                                                                     Similar clone fragments
 except for layout, white space, and
                                                     except for literals, identifiers, data types,
                                                                                                               with added, changed, and removed
                                                       layout, white space, and comments.
                                                                                                                        some statements.
               comments
         14HOK 100%
                                                                                                                    องค์ประกอบเนบอนกัน เท่านั้น
```

```
private static String getFormatByName(String name){
   if(name != null){
        final int j = name.lastIndexOf(".") + 1,
        k = name.lastIndexOf("/") + 1;
        if(j > k &&
        j < name.length()){
            return name.substring(j);
        }
    }
   return null
}

Code fragments that have the same computation
   but different in syntax or algorithm.</pre>

public static String getExtension(final String filename){
        if(filename == null ||
        filename.trim().length == 0 ||
        if(filename.enull ||
        filename.trim().length == 0 ||
        if(filename == null ||
        filename.trim().length == 0 ||
        return null;
   }
   }
   int pos = filename.substring(pos+1);
   }
}
```

คำในต่องกับ Code detector: บรินารจัดการโดก กานในกู้ได้
กับการขึ้นย Code สำหรับ Code ก็ private ไว้
Process:



Model: Merry
· Wah Al Mide Web - based

· feature คราวตามสนให้างลาง

· HEN type 1, type 3

Merry: Web-Based Code Clone Detection System Using Machine Learning

Features	Machine Learning	BigCloneBench
11 Syntactic Code Metrics + + 12 Semantic Code Metrics (code2vec)	Decision tree, Random forest, SVM, SVM + SMO	Largest and Credible clone data
Web-based Tool with User Interface	GitHub Integration	Visualizations and reports

Data collection

Or Big Clone Bench Dataset Flidh Script ANDROYN

Training Data [8]						
	Tru					
Type 1	Type 2	Very Strong Type 3	Strongly Type 3	False Clone Pairs		
13,750	3,104	1,207	4,602	22,663		

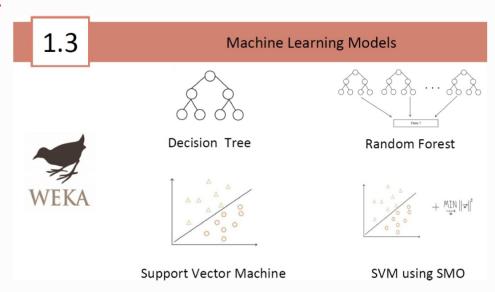
Testing Data

	True Clone Pa	Falsa Clava Daire			
Type 1	Type 2	Very Strong Type 3	Strongly Type 3	False Clone Pairs	
2,383	557	307	1,477	18,893	

Code metrics Extraction

- · Syntactic metrics : ๑ โดรมสร้างการเขียน
- · Semantic metrics: ansinson Pr Code 2 vec ille Feature extractor

· Model



· Model Evaluation

Model	Metrics	Precision	Recall	F1-Score
Randomization (baseline)		0.20	0.49	0.28
	Syntactic + Semantic	0.89	0.86	0.87
Decision Tree	Syntactic	0.95	0.72	0.86
	Semantic	0.68	0.87	0.76
	Syntactic + Semantic	0.97	0.86	0.91
Random Forest	Syntactic	0.97	0.80	0.87
	Semantic	0.70	0.87	0.78
	Syntactic + Semantic	0.97	0.85	0.91
SVM	Syntactic	0.97	0.79	0.87
	Semantic	0.62	0.90	0.73
	Syntactic + Semantic	0.98	0.89	0.93
SVM using SMO	Syntactic	0.97	0.69	0.81
	Semantic	0.63	0.90	0.74