No one can cheat the time, no matter how smart I am, it still very far until I understand the natural of intelligent, and understand at least at a surface, what is god, why gods create human, and who created gods? Did it intentionally, or just purely coincidental.

What is the natural of intelligent?
Does machine intelligence work the same like us?.
Does machine learning is just stay in the black box, that we never actually understand it? Wait, but if we cannot truly understand it, how dare we sure they will produce the correct solutions?

Let pick a random, but somewhat useful, practical, and latest AI problem in Kaggle.

In[237]:=

```
SetDirectory["~/nhannht-projects/AI"]
```

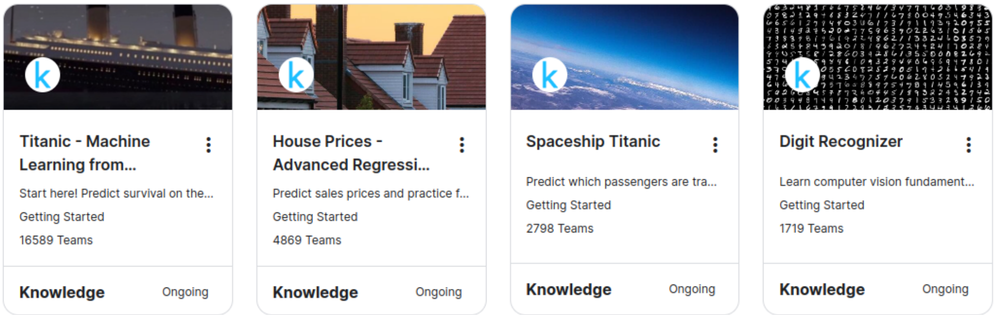
Out[237]=

```
/home/vermin/nhannht-projects/AI
```

Competitions

There is 10 indefinitely competitions (a hackathon that never end), oh we see, despite call themself hackathon, it actually like a collection of challenges just for learning. At least if I finished those competitions, and the projects from AWS “scholarships”, maybe I can do actually know a bit about AI

In[9]:=



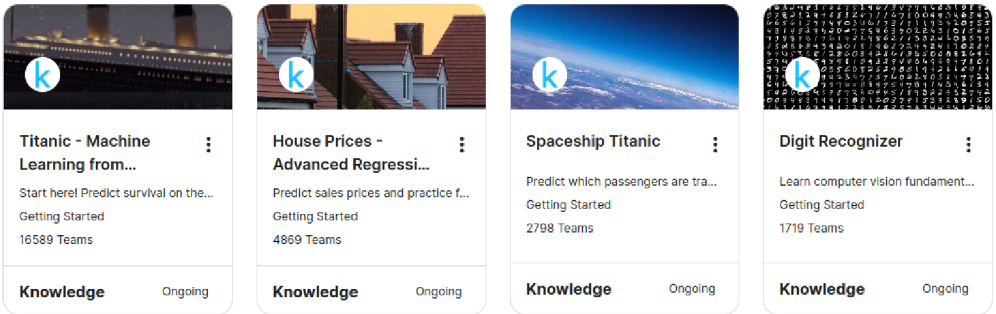
Titanic - Machine Learning from...
Start here! Predict survival on the...
Getting Started
16589 Teams
Knowledge Ongoing

House Prices - Advanced Regressi...
Predict sales prices and practice f...
Getting Started
4869 Teams
Knowledge Ongoing

Spaceship Titanic
Predict which passengers are tra...
Getting Started
2798 Teams
Knowledge Ongoing

Digit Recognizer
Learn computer vision fundament...
Getting Started
1719 Teams
Knowledge Ongoing

Out[9]=



Titanic - Machine Learning from...
Start here! Predict survival on the...
Getting Started
16589 Teams
Knowledge Ongoing

House Prices - Advanced Regressi...
Predict sales prices and practice f...
Getting Started
4869 Teams
Knowledge Ongoing

Spaceship Titanic
Predict which passengers are tra...
Getting Started
2798 Teams
Knowledge Ongoing

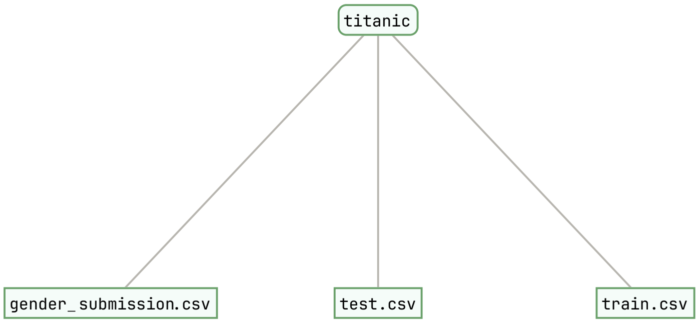
Digit Recognizer
Learn computer vision fundament...
Getting Started
1719 Teams
Knowledge Ongoing

Titanic

In[233]:=

```
FileNameJoin[{Directory[], "data/titanic"}] // FileSystemTree
```

Out[233]=



In[238]:=

```
dataFiles = FileNames[All, FileNameJoin[{Directory[], "data/titanic"}]];
dataFiles // Column
```

Out[239]=

```
/home/vermin/nhannht-projects/AI/data/titanic/gender_submission.csv
/home/vermin/nhannht-projects/AI/data/titanic/test.csv
/home/vermin/nhannht-projects/AI/data/titanic/train.csv
```

In[16]:=

```
importCSVAsDataset[file_String] := Import[file, "Dataset", HeaderLines → 1];
```

In[17]:=
```
trained =
  importCSVAsDataset["/home/vermin/nhannht-projects/AI/data/titanic/train.csv"]
```

Out[17]=

PassengerId	Survived	Pclass	Name
1	0	3	Braund, Mr. Owen Harris
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)
3	1	3	Heikkinen, Miss. Laina
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)
5	0	3	Allen, Mr. William Henry
6	0	3	Moran, Mr. James
7	0	1	McCarthy, Mr. Timothy J
8	0	3	Palsson, Master. Gosta Leonard
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)
11	1	3	Sandstrom, Miss. Marguerite Rut
12	1	1	Bonnell, Miss. Elizabeth
13	0	3	Saundercock, Mr. William Henry
14	0	3	Andersson, Mr. Anders Johan
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina
16	1	2	Hewlett, Mrs. (Mary D Kingcome)
17	0	3	Rice, Master. Eugene
18	1	2	Williams, Mr. Charles Eugene
19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)
20	1	3	Masselmani, Mrs. Fatima

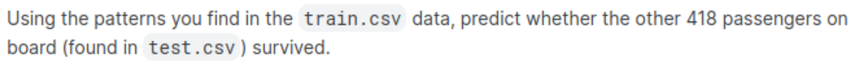
rows 1–20 of **891** | columns 1–10 of **12**

In[18]:=
```
tests =
   importCSVAsDataset["/home/vermin/nhannht-projects/AI/data/titanic/test.csv"];
```

In[19]:=
```
trained[1] // Normal
```

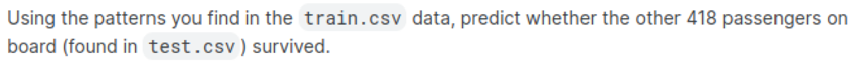
Out[19]=
```
<|PassengerId → 1, Survived → 0, Pclass → 3,
 Name → Braund, Mr. Owen Harris, Sex → male, Age → 22, SibSp → 1,
 Parch → 0, Ticket → A/5 21171, Fare → 7.25, Cabin → , Embarked → S|>
```

In[20]:=



Using the patterns you find in the `train.csv` data, predict whether the other 418 passengers on board (found in `test.csv`) survived.

Out[20]=



Using the patterns you find in the `train.csv` data, predict whether the other 418 passengers on board (found in `test.csv`) survived.

In[47]:=
```
trainedHeader = tests[1, Keys] // Normal
```

Out[47]=
```
{PassengerId, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked}
```

First, we must extract the column that will affect the outcome, purely based on the our logic. Example PassengerID, SibSpParch ,Name is no necessary here.

In[234]:=
```
trained1 =
 trained[All, <|"Survived" → # Survived, "Pclass" → # Pclass, "Sex" → # Sex,
    "Age" → # Age, "Fare" → # Fare, "Cabin" → # Cabin|> &]
```

Out[234]=

Survived	Pclass	Sex	Age	Fare	Cabin
0	3	male	22	7.25	
1	1	female	38	71.2833	C85
1	3	female	26	7.925	
1	1	female	35	53.1	C123
0	3	male	35	8.05	
0	3	male		8.4583	
0	1	male	54	51.8625	E46
0	3	male	2	21.075	
1	3	female	27	11.1333	
1	2	female	14	30.0708	

rows 1–10 of 891

Second, we should delete the column that have missing data so much

```
In[61]:= <|# → trained1[Count[""], #] / Length[trained1]|> & /@ trained1[1, Keys]
Out[61]=
```

Survived	0
Pclass	0
Sex	0
Age	0.198653
Fare	0
Cabin	0.771044

The cabin column have 77 percent missing data, better remove it

```
In[199]:=

trained1 = trained[All, <|"Survived" → # Survived,
    "Pclass" → # Pclass, "Sex" → # Sex, "Age" → # Age, "Fare" → # Fare|> &]
Out[199]=
```

Survived	Pclass	Sex	Age	Fare
0	2	male	59	13.5
1	3	female	5	31.3875
0	2	male	24	10.5
0	3	female		7.55
0	2	male	44	26

rows 233–237 of **891**

```
In[88]:= trained1[Select[# Age ≠ "" &], "Age"] // Mean
Out[88]=
29.6991
```

Well, let trained using super function Classify from wolfram

What about age, it have only 20 percent data missing, better replace those missing by mean values

In[235]:=

```
trained2 =
 trained1[All, {"Age" → Replace["" → (trained1[Select[# Age≠ "" &], "Age"] // Mean)]}]
```

Out[235]=

Survived	Pclass	Sex	Age	Fare	Cabin
0	3	male	22	7.25	
1	1	female	38	71.2833	C85
1	3	female	26	7.925	
1	1	female	35	53.1	C123
0	3	male	35	8.05	
0	3	male	29.6991	8.4583	
0	1	male	54	51.8625	E46
0	3	male	2	21.075	
1	3	female	27	11.1333	
1	2	female	14	30.0708	

rows 1–10 of **891**

In[213]:=

```
classifier = Classify[trained2 → "Survived"]
```

Out[213]=

```
ClassifierFunction[ Input type: Mixed (number: 4)
                    Classes: 0, 1 ]
```

In[112]:=

```
classifer@ tests⟦20⟧ // Quiet
```

Out[112]=

```
1
```

In[169]:=

```
tests // Length
```

Out[169]=

```
418
```

In[183]:=

```
tests1 =
  tests[All, <|"Pclass" → #Pclass, "Sex" → #Sex, "Age" → #Age, "Fare" → #Fare|> &];
tests2 =
 tests1[All, {"Age" → Replace["" → (tests1[Select[#Age ≠ "" &], "Age"] // Mean)],
   "Fare" → Replace["" → (tests1[Select[#Fare ≠ "" &], "Fare"] // Mean)]


  }]
```

Out[184]=

Pclass	Sex	Age	Fare
3	male	34.5	7.8292
3	female	47	7
2	male	62	9.6875
3	male	27	8.6625
3	female	22	12.2875

rows 1–5 of **418**

In[227]:=

```
testsWithSurvived = tests2[All, <|#, "Survived" → classifier[#]|> &]
```

Out[227]=

Pclass	Sex	Age	Fare	Survived
3	male	34.5	7.8292	0
3	female	47	7	0
2	male	62	9.6875	0
3	male	27	8.6625	0
3	female	22	12.2875	0
3	male	14	9.225	0
3	female	30	7.6292	1
2	male	26	29	0
3	female	18	7.2292	1
3	male	21	24.15	0

rows 1–10 of **418**

Well, it ... done if we consider the requirements in Kaggle, just push the dataset as a results .

It is time to dive deep. I use super function Classify, the super meaning that this function give me a very high level of abstraction, it automatic pick from variable types to the algorihms to implement AI pipe line. IF you consider it is cheat, well, i am sure a ton of ML library in Python nowadays using the same style. People nowadays learn AI without actually know what happen under the iceberg.

Using the Information function, we can see classifier show it accuracy rate is from 72->80%. It pretty low, but still better than 50-50 purely random pick. IT still low because of the low number of training set. But, we know, no matter how big our training set is, the is a upper limit of how accurate AI can predict, and it will never 100% percent.   . It is using RandomForest method. Well, the Wolfram super function smart enough it automatic pick the algorithm on it own.

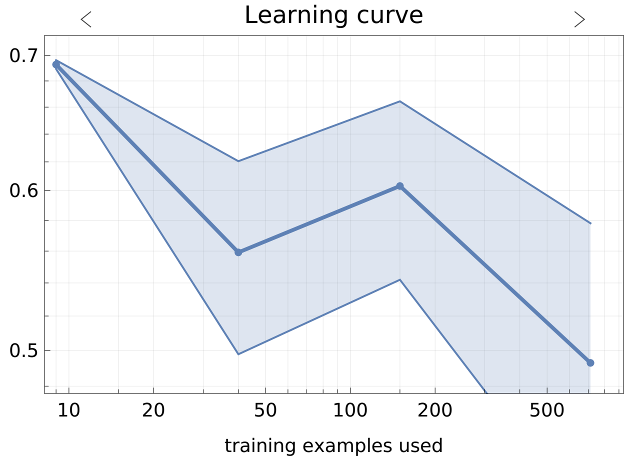
In[216]:=

```
Information[classifier]
```

Out[216]=

Classifier information

Data type	Mixed (number: 4)
Classes	0, 1
Accuracy	(79. ± 4.)%
Method	NearestNeighbors
Single evaluation time	4.72 ms/example
Batch evaluation speed	26.8 examples/ms
Loss	0.493 ± 0.085
Model memory	355. kB
Training examples used	891 examples
Training time	2.73 s



```
In[218]:=
<|# → Information[classifier, #]|> & /@
  Information[classifier, "Properties"] // Dataset
Out[218]=
```

MeanCrossEntropy	0.49 ±0.09
Method	NearestNeighbors
MethodDescription	The nearest neighbors predictor predicts the value of a new exa
MethodOption	Method → {"NearestNeighbors", "Ne
MethodParameters	<|NeighborsNumber → 20, DistributionSmoot
MissingSynthesizer	...
PerformanceGoal	Automatic
Properties	{ ⋯31 }
TrainingClassPriors	<|0 → 0.615901, 1 → 0.384099|>
TrainingTime	2.73487 s

rows 21–30 of **31**

```
In[224]:=
Divide @@ trained2[Counts, "Survived"] // N
Out[224]=
1.60526
In[229]:=
Divide @@ testsWithSurvived[Counts, "Survived"] // N
Out[229]=
1.84354
```

Well, temporally stop here. Still so shallow and full of darkness. I need sharpen the knowledge in some related fields

Scratchpad

```
In[29]:= SetDirectory["~/nhannht-projects/AI/"];
In[30]:= NotebookSave[EvaluationNotebook[], FileNameJoin[{Directory[], "basic.nb"}]]
In[242]:=
VerminExportKeepSyntaxHighLight[]
```

```
In[245]:=
Export[FileNameJoin[{Directory[], "basic.pdf"}], EvaluationNotebook[]]

Export: First argument NotebookObject[ basic.nb ] is not a valid file specification.

Out[244]=
$Failed
```