

Basic Machine Learning with R

MACHINE LEARNING with kaggle™

WORKSHOP

19/02/2018

by Data Science ເຂົາເຈົ້າ & Data Rockie

INTRO TO KAGGLE



The popular platform for
data science competitions

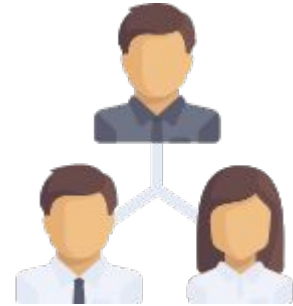
Data Science Competitions



Prize Money

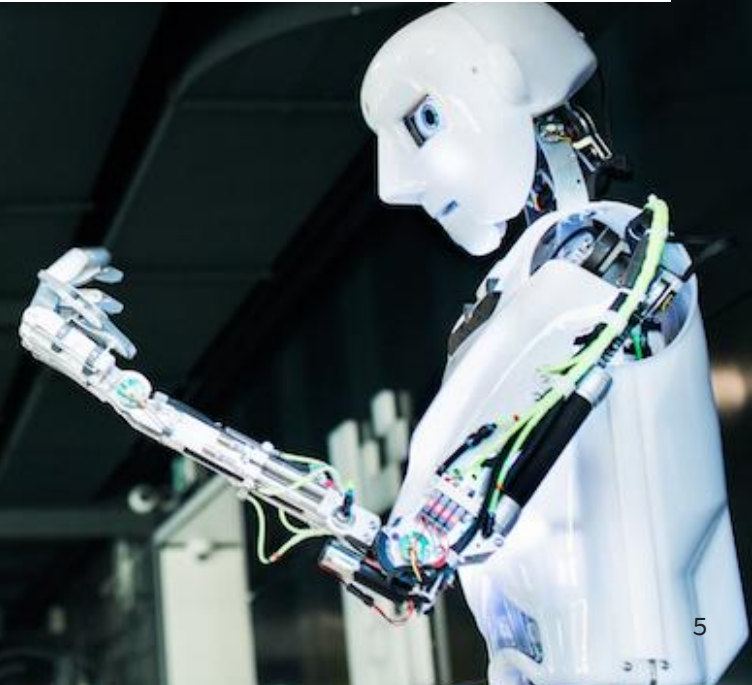


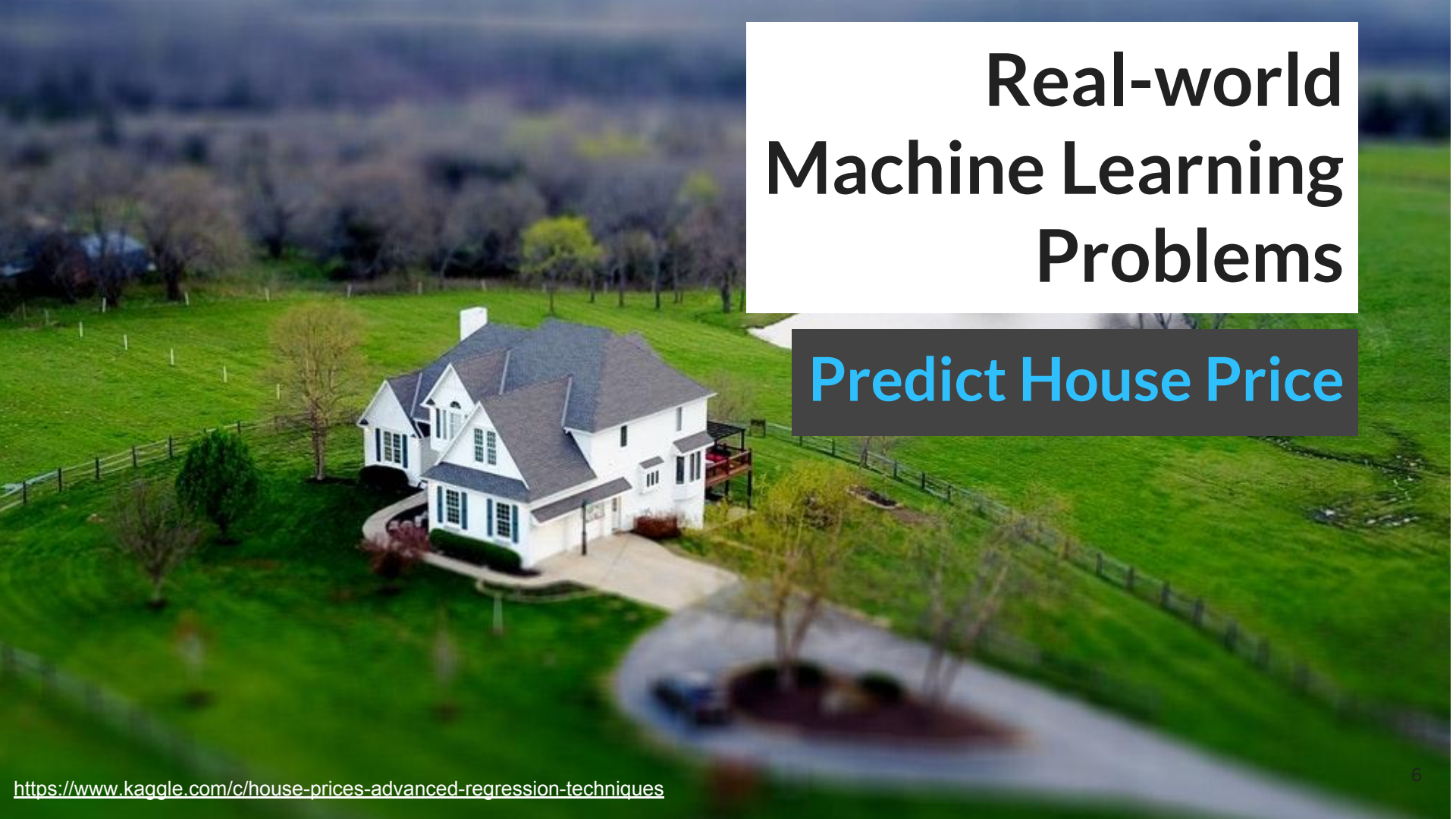
Time Limit



Teamwork

Real-world Machine Learning Problems





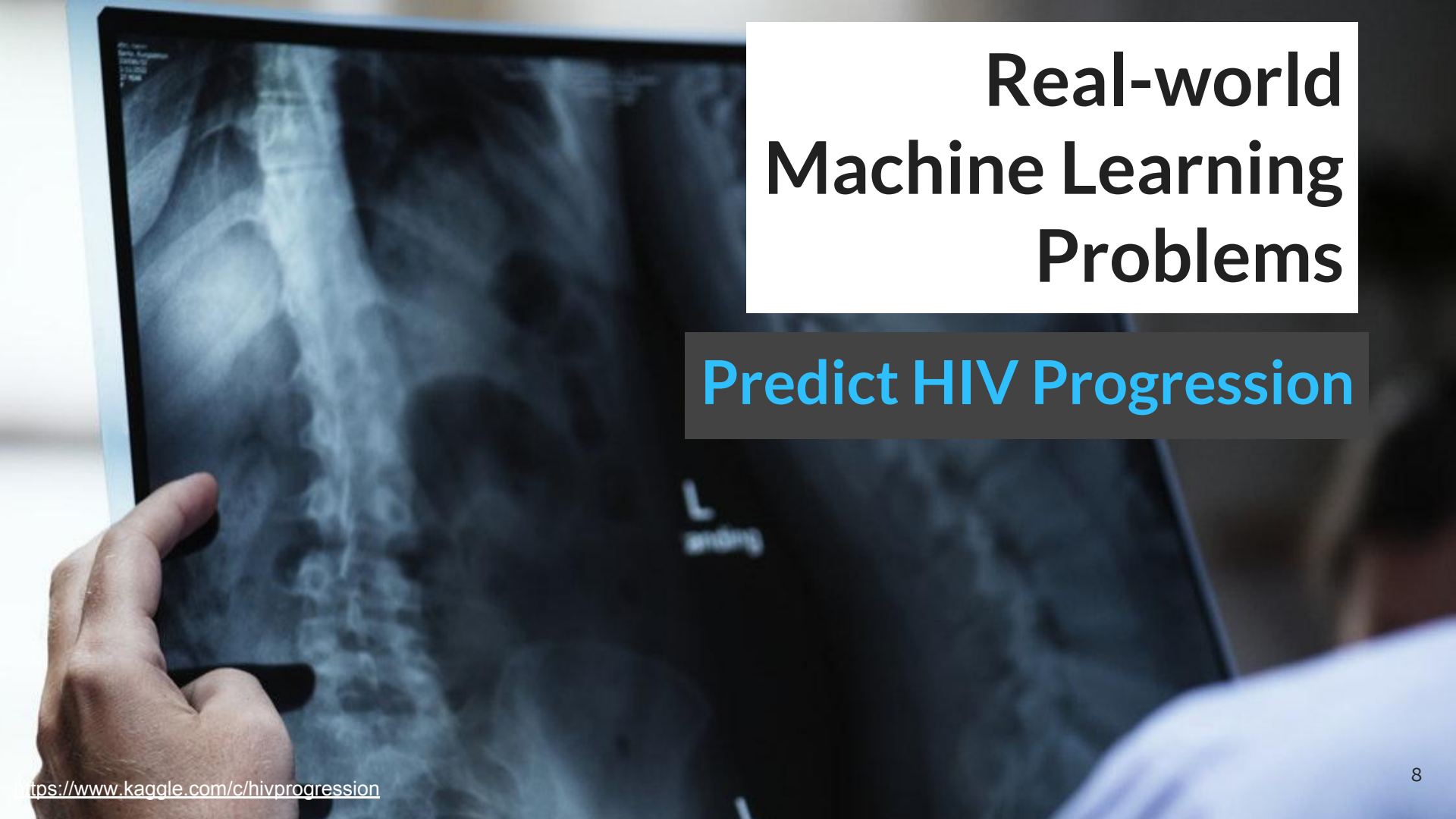
Real-world Machine Learning Problems

Predict House Price

Real-world Machine Learning Problems


Revenue Forecasting





Real-world Machine Learning Problems

Predict HIV Progression

A satellite image of a volcanic landscape. In the upper right, there is a dark, irregularly shaped crater lake. Below it, a large, dark, and textured area represents a lava flow that has advanced from the top center towards the bottom left. The surrounding terrain is rugged and grey, with patches of green vegetation. A prominent, light-colored, linear feature, possibly a road or a dry riverbed, runs vertically through the center of the image. The overall scene is a high-contrast, grayscale-like image with some color in the vegetation and the lake.

Real-world Machine Learning Problems

Classify Satellite Images

Acquired by Google in 2017

Google Cloud kaggle



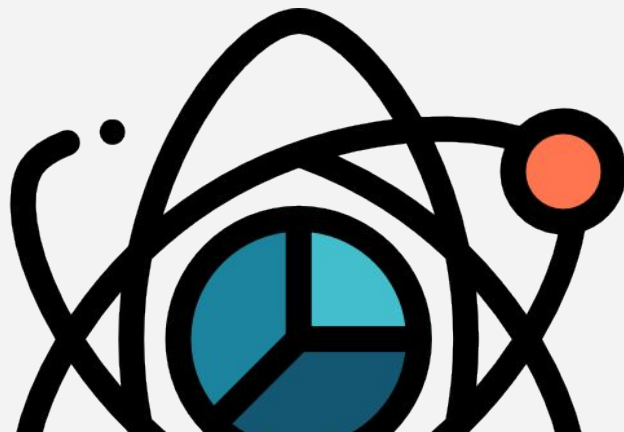
Is Kaggle a **good way** to become **data scientist**?

YES

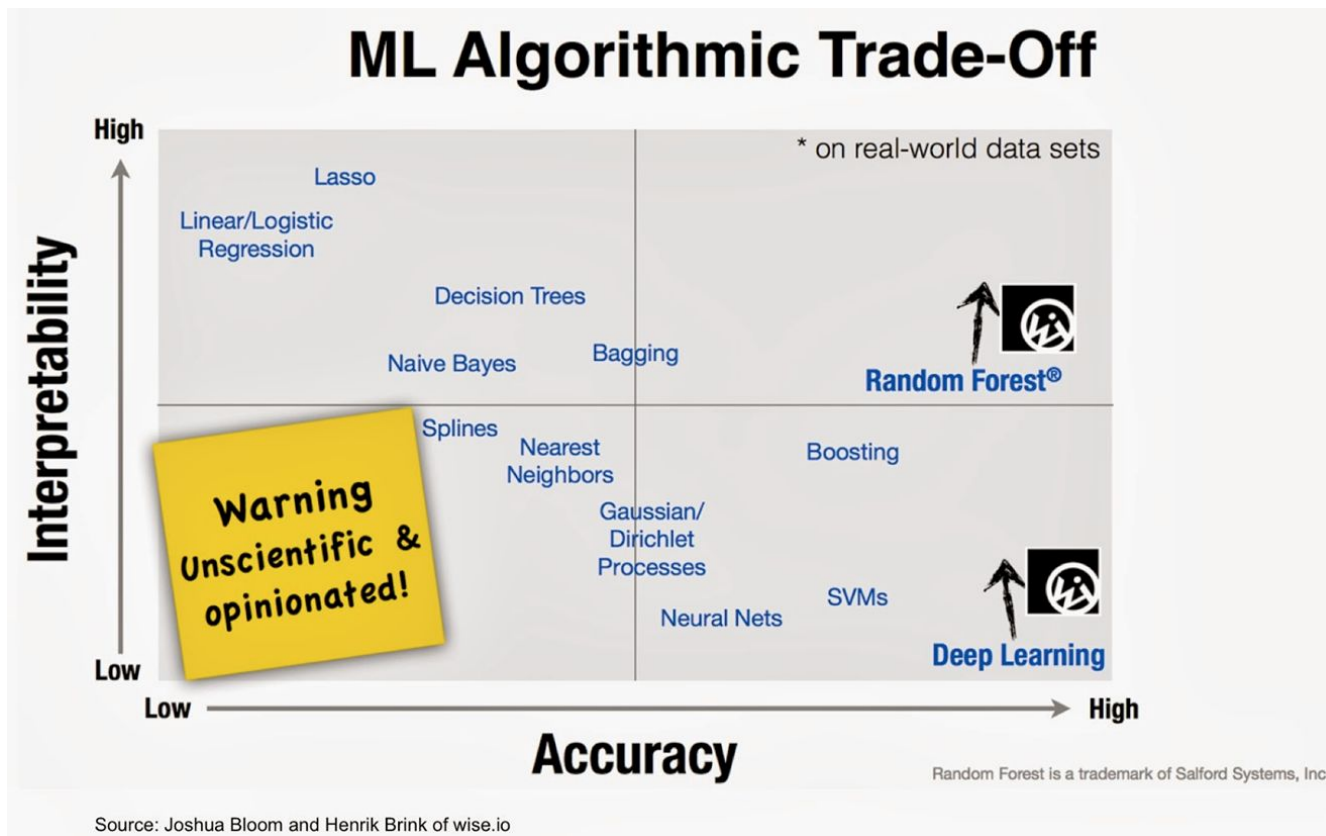
- Real-world problem, data
- Learn new library in R or Python
- Learn from many people & winners sharing their explorations / solutions.

NO

- Skip data collection & cleaning
- New library may not be practical



Model accuracy & interpretability Tradeoff



Interpretability
Easy to understand

VS

Accuracy
Easy to get high prediction score

“As long as complex models are properly validated, it may be improper to use a model that is built for interpretation rather than predictive performance.”

- Kuhn and Johnson “Applied Predictive Modeling”

Accuracy > Interpretability



Get started with Kaggle

What Kaggle problem looks like?

Example: Predict who will like ice cream?

Name	Age	Weight (KG)	Like Ice Cream?
Chatri	24	60	Yes
Tong	30	50	No
Somsri	42	48	No
Thanet	18	72	We have to predict the answers
Petch	35	48	
Pongsak	26	62	

Data Science Process

“Turn data into insights”

Collect Data > Maintain >
Explore Data > Data Cleaning > Build Model >
Visualize > Present

Kaggle Process



“Turn data into prediction”

~~Collect Data~~ > ~~Maintain~~ >
Explore Data > Data Cleaning > Build Model >
~~Visualize~~ > ~~Present~~

**“No matter how advanced is your Machine Learning algorithm,
the results will be bad if the input data is bad.”**

- *Kaiser Fung “Numbersense: how to use Big Data to your advantage”*



WORKSHOP OVERVIEW

What will we do today?

Kaggle competitions for beginners

- No time limit
- Public leaderboard reset every 2 months

2 Workshops

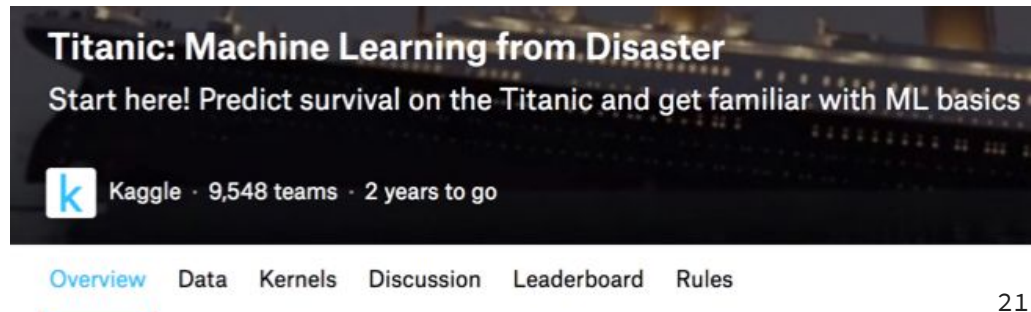
1. **Classification Problem:** Who is likely to survive on Titanic?
2. **Regression Problem:** predict the house price from its quality

Workshop 1: Titanic

Kaggle Page: <https://www.kaggle.com/c/titanic>

Kaggle Project Tabs

1. **Overview** (description, goal, evaluation metric, submission format)
2. **Data** (Test, Train, Data dictionary)
3. **Kernels** (Learn from others)
4. **Discussion** (Webboard)
5. **Leaderboard** (Public & Private)
6. **Rules** (Timeline)



Many **Features** & **1 Target Variable**

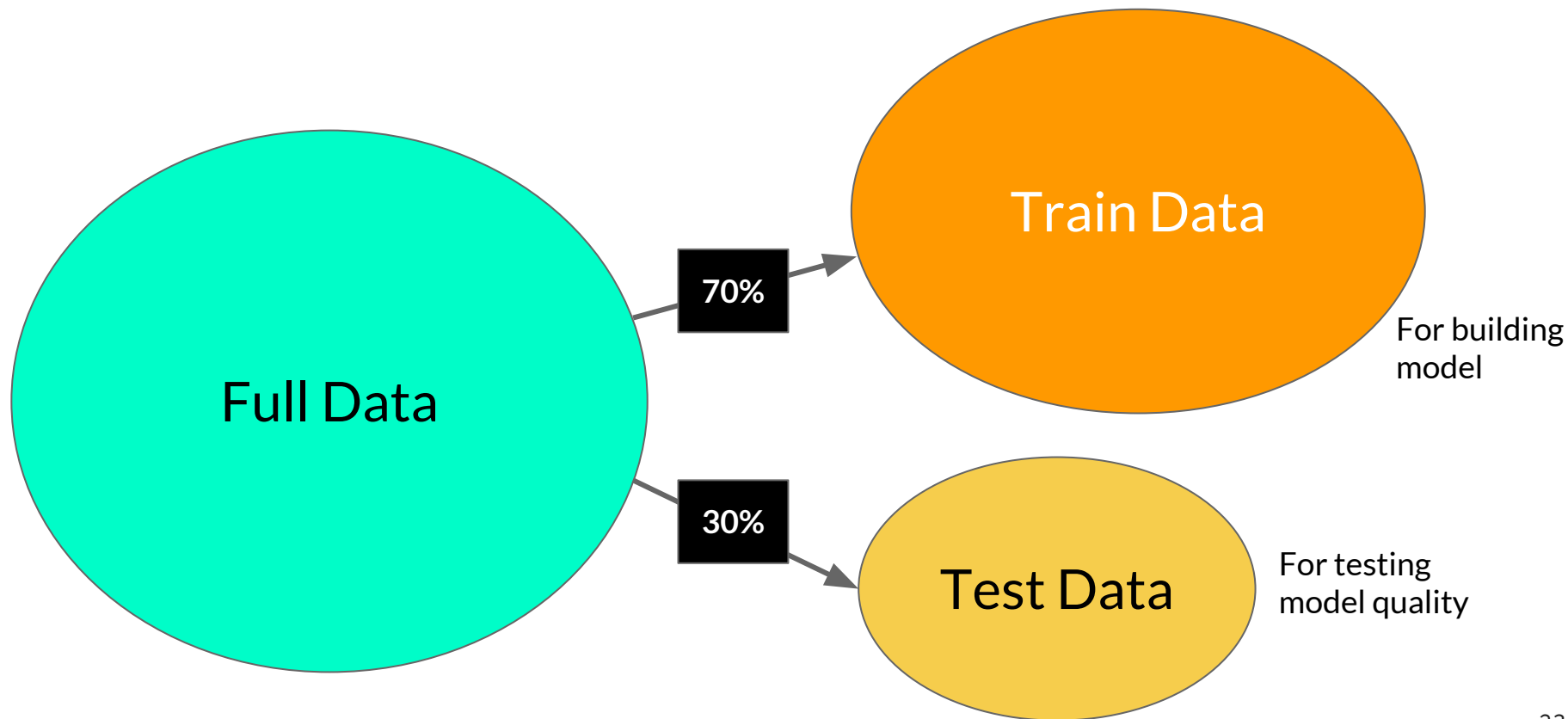
Example: Predict who will like ice cream?

Name	Age	Weight (KG)	Like Ice Cream?
Chatri	24	60	Yes
Tong	30	50	No
Somsri	42	48	No
Thanet	18	72	We have to predict the answers
Petch	35	48	
Pongsak	26	62	

X: Features

Y: Target Variable


Data needs to be splitted




What is **train** & **test** data

Example: Predict who will like ice cream?

Name	Age	Weight (KG)	Like Ice Cream?
Chatri	24	60	Yes
Tong	30	50	No
Somsri	42	48	No
Thanet	18	72	We have to predict the answers
Petch	35	48	
Pongsak	26	62	

 Train

 Test

What is Public Leaderboard & Private Leaderboard

Test Data

Name	Age	Weight (KG)	Like Ice Cream?
Thanet	18	72	We have to predict the answers
Petch	35	48	
Pongsak	26	62	
Kate	16	42	

Public

Private

WORKSHOP 1

Workshop 1: Titanic

Question: Who will survive from Titanic?

Step-by-step guide:

1. Load data
2. Explore data
3. Data cleaning
4. Model training





Titanic

The ship from England to United States in 1912

A photograph of an iceberg floating in the ocean. The tip of the iceberg is visible above the water line, while the much larger submerged portion is visible below. White brackets on the right side of the image indicate the proportions: a small bracket for the tip labeled '10%' and a large bracket for the submerged part labeled '90%'. The text 'Tip of the iceberg' is written in white on the left side, spanning the water line.

Tip of the iceberg

10%

90%

The first rule of Kaggle club:
You don't talk about Prediction
(without data dictionary)





First step of ML:
Find the [target variable](#)

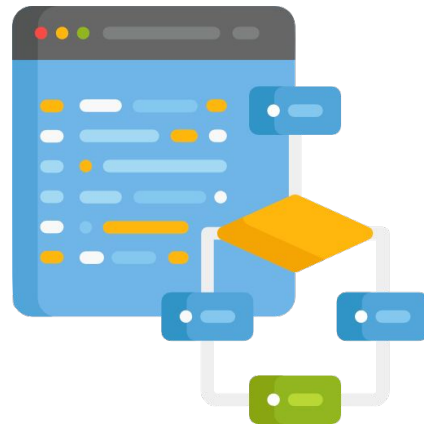
Titanic: Data Dictionary

Variable	Definition	Key
survival	(Target Variable) Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Let's get to the code

Structure of “workshop1-titanic.R”

1. Load data
2. Explore data
 - a. Summary
 - b. Correlation
3. Clean data
 - a. Impute missing values
 - b. Convert data types
4. Train model using **decision tree** & Visualize
5. Train model with **random forest** & optimize using **random search**
6. Use model to predict the test data
7. Export file to submit to Kaggle



Titanic: Explore Data

Titanic Fact - <https://www.encyclopedia-titanica.org/>

Find answers from the data using R:

1. How many passengers?
2. How many people in each ticket class?
3. What is the average age of people on Titanic?
4. What is the correlation between variables?

Titanic: Cleaning Data

3 Types of Data Anomalies

1. **Missing Values** (where there should be data) - delete or impute
2. **Wrong Data** (misspelling, wrong column)
3. **Incomplete Data** (abbreviation, multiple measurement unit, incomplete address)

Titanic: Evaluation Metric = Accuracy

Confusion matrix →

n=165		Predicted: NO	Predicted: YES
Actual: NO	50	10	
Actual: YES	5	100	

■ Correct answers

Correctly predict 0% → Accuracy = 0

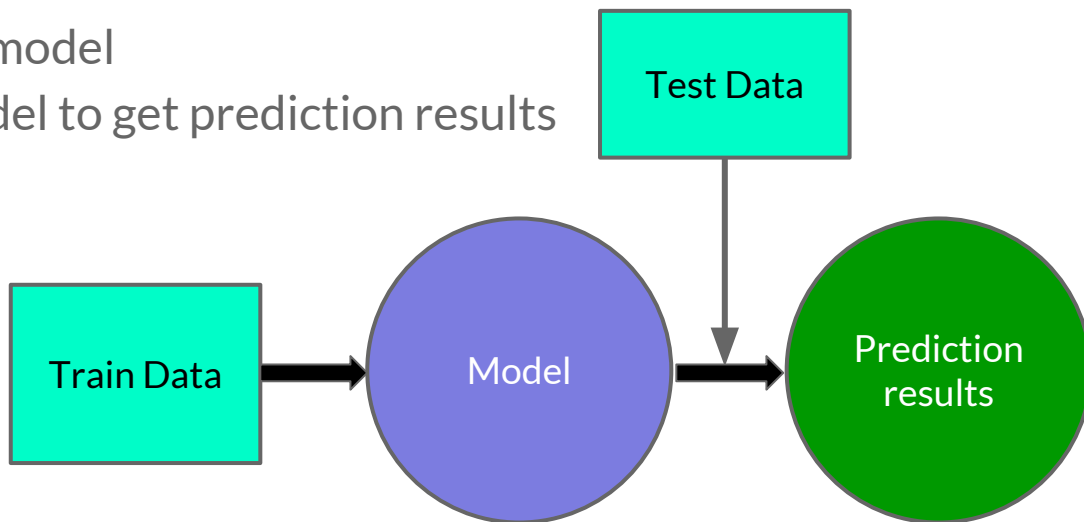
Correctly predict 50% → Accuracy = 0.5

Correctly predict 100% → Accuracy = 1

Titanic: Building Model

How to build a prediction model

1. Prepare features and target variables
2. Use train data to train the model
3. Feed test data into the model to get prediction results
4. Optimize the model
5. Submit to [Kaggle](#)



Titanic: Feature Engineering

Feature Engineering = Build a feature from the current attributes

e.g. Turn numeric into categorical --> Price (\$800, \$5000, \$8000, ...) to
Price_less_than_5000, Price_more_than_5000

e.g. Extract some parts --> Date (30-5-2017, ...) to
Hour_of_day, Day_or_night

e.g. One Hot Encoding --> Color (red, blue, green) to
Is_red, Is_blue, Is_green

Titanic: Optimize the model with Grid Search

What is the easiest way to find the best parameter?

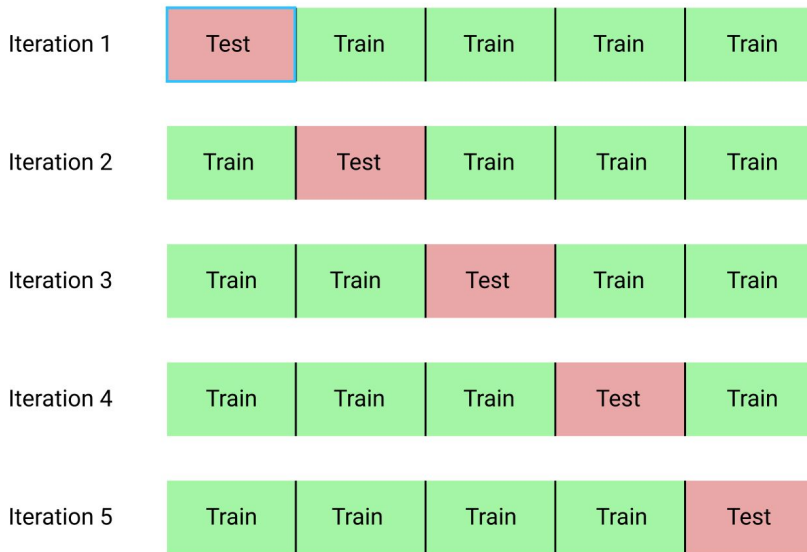


```
gbmGrid <- expand.grid(interaction.depth = c(1, 5, 9),  
                        n.trees = (1:30)*50,  
                        shrinkage = 0.1,  
                        n.minobsinnode = 20)  
  
gbmFit2 <- train(Class ~ ., data = training,  
                 method = "gbm",  
                 trControl = fitControl,  
                 verbose = FALSE,  
                 ## Now specify the exact models  
                 ## to evaluate:  
                 tuneGrid = gbmGrid)
```


Titanic: Evaluate the model before submission

K-Cross Validation

Split data into k equal portions. Use 1 portion for test, and the rest for train for k times.



K = 5

People normally
use 5 or 10


Titanic: How to submit

The screenshot shows the top navigation bar with links: Overview, Data, Kernels, Discussion, Leaderboard, and Rules. A red box highlights the 'Join Competition' link in the top right corner. Below the navigation bar, a grey banner states: 'You have 9 submissions remaining today. This resets 14 hours from now (00: 00 UTC)'. The number '9' is highlighted with a red box. The main content area is titled 'Step 1 Upload submission file'. It features a large dashed box containing an upload icon (an arrow pointing up from a document) and the text 'Upload Submission File'. Below this box, there are two columns of text. The left column is titled 'File Format' and states: 'Your submission should be in CSV format. You can upload this in a zip/gz/rar/7z archive, if you prefer.' The right column is titled 'Number of Predictions' and states: 'We expect the solution file to have 418 prediction rows. This file should have a header row. Please see sample submission file on the [data page](#).'

Overview Data Kernels Discussion Leaderboard Rules [Join Competition](#)

You have 9 submissions remaining today. This resets 14 hours from now (00: 00 UTC).

Step 1
Upload submission file


Upload Submission File

File Format
Your submission should be in CSV format.
You can upload this in a zip/gz/rar/7z archive, if you prefer.

Number of Predictions
We expect the solution file to have 418 prediction rows. This file should have a header row. Please see sample submission file on the [data page](#).

Warning: There is a [limit](#) you can submit per day.

WORKSHOP 2

Workshop 2: House Price Prediction

Kaggle Page: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques>

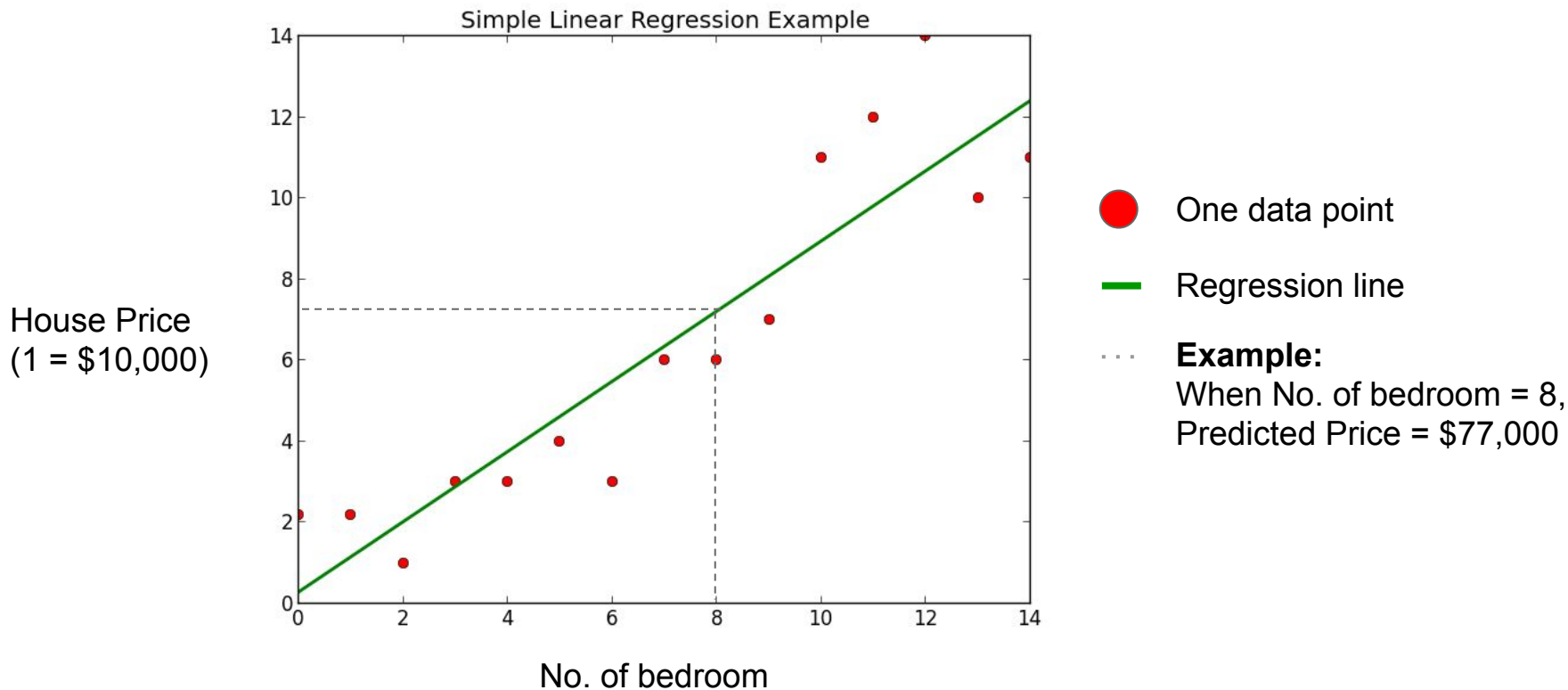
Problem type: Regression

Regression VS Classification problem?

Regression: Target variable = continuous number (e.g. 1, 1.2, 1.5)

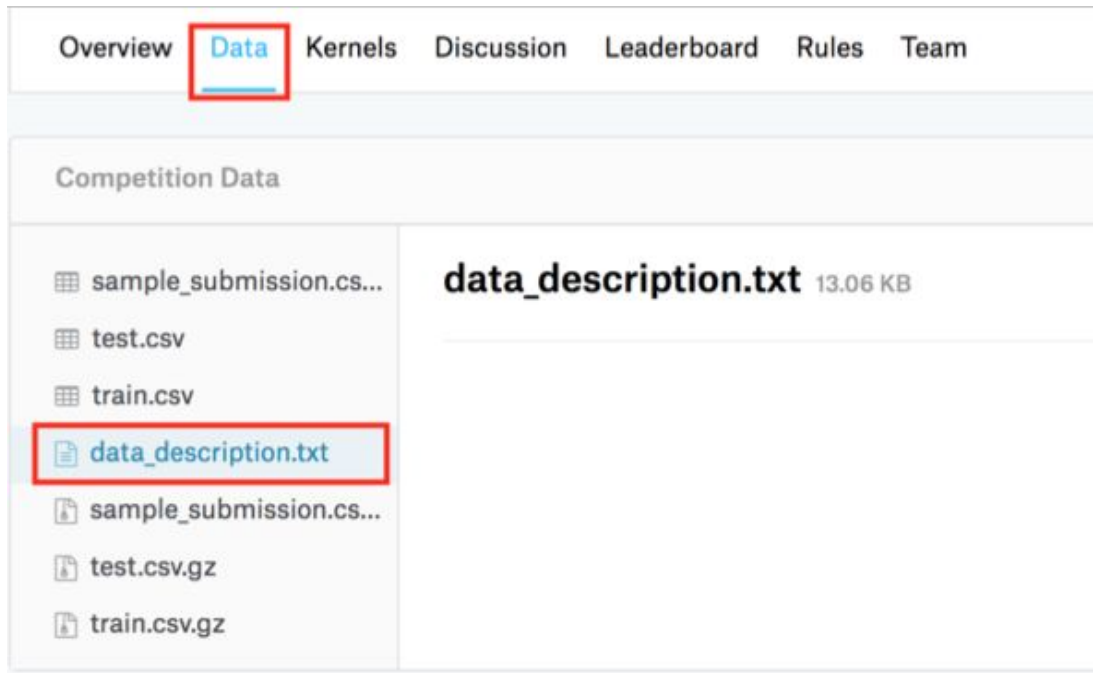
Classification: Target variable = categorical (e.g. Yes = 1, No = 0)

Regression Problem



Workshop 2: House Price Prediction

Data Dictionary can be found in: [Data > data_description.txt](#)



Workshop 2: House Price Prediction

Question: What is the house sale price from its quality?

Step-by-step guide:

1. Load data
2. Explore data
3. Data cleaning
4. Model training



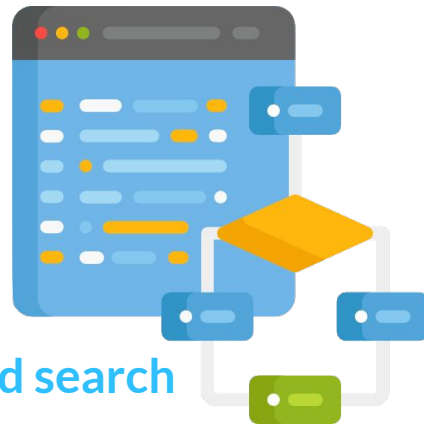


First step of ML:
Find the [target variable](#)

Let's get to the code

Structure of “workshop2-houseprice.R”

1. Load data
2. Explore data
 - a. Summary
 - b. Correlation
3. Clean data
 - a. (Add any function you would like)
4. Train model using **random forest** & optimize using **grid search**
5. Use model to predict the test data
6. Export file to submit to Kaggle



House Price: Evaluation Metric = RMSE

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum (\hat{Y}_i - Y_i)^2}$$

Sum (for each data point):
N = No. of data point
 \hat{Y} = Predicted price
Y = Real price

RMSE = How big is the prediction error?

More RMSE = less accurate model

RMSE 0 = No difference

Less RMSE = more accurate model

How to get the most of Kaggle?

Learn from the “Most Votes” Kernel

The screenshot shows the Kaggle Kernels interface. The top navigation bar includes 'Overview', 'Data', 'Kernels' (highlighted with a red box), 'Discussion', 'Leaderboard', and 'Rules'. A red arrow points from the 'Kernels' tab to the 'Most Votes' sort option in the 'Sort by' dropdown menu. The 'Sort by' menu also includes 'Hotness', 'Most Comments', 'Best Score', 'Recently Created', and 'Recently Run'. The main content area displays a list of kernels, each with a rank, a user profile picture, a title, a timestamp, a score, and tags. The kernels are sorted by 'Most Votes'.

Rank	User	Kernel Title	Time Ago	Score	Tags	Visuals	Py	Comments
2075		Exploring Survival on the Titanic	1mo ago	0.80382	tutorial, beginner, feature engineering, random forest			
1221		Introduction to Ensembling/Stacking in Python	21d ago		tutorial, ensembling, xgboost			
851		Titanic Data Science Solutions	1mo ago		tutorial, feature engineering, model comparison		Py	350
809		A Journey through Titanic	1y ago	0.74162	beginner, eda, random forest, logistic regression		Py	351
473		EDA To Prediction(DieTanic)	21d ago		eda, data visualization, classification, ensembling, model comparison		Py	84



Participate in the forum discussion

[Overview](#) [Data](#) [Kernels](#) [Discussion](#) [Leaderboard](#) [Rules](#) [New Topic](#)

1,635 topics and kernels [Subscribe](#) Sort by **Hotness**

[All](#) [Mine](#) | [Upvoted](#) Topics & Kernels Search topics

80





Rolling Leaderboards
[William Cukierski](#) 4 years ago

last comment by
[Shuang Mo](#) 9mo ago

44

80





End to End Project with Python
[Niklas Donges](#) last run 9 days ago

last comment by
[Manuel Machado](#) 17h ago

48

1





Anyone need a team member?
[Mikel Kengni](#) 3 hours ago

last comment by
[VinayakPahalwan](#) 1h ago

1

0





First try with random forest
[Kyon Huang](#) 6 hours ago

last comment by
[Kyon Huang](#) 6h ago

0

0



Titanic: Error, unexpected symbol
[lisamar](#) a day ago

last comment by
[lisamar](#) 1d ago

0

Invite friends to do Kaggle together

Facebook Group:

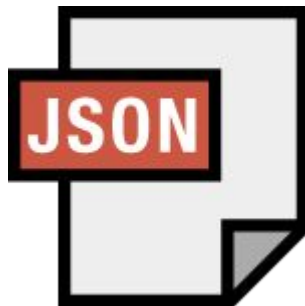
Thai Data Scientists & Kagglers:

<https://www.facebook.com/groups/thaidsml>

We will discuss & update in this group :)

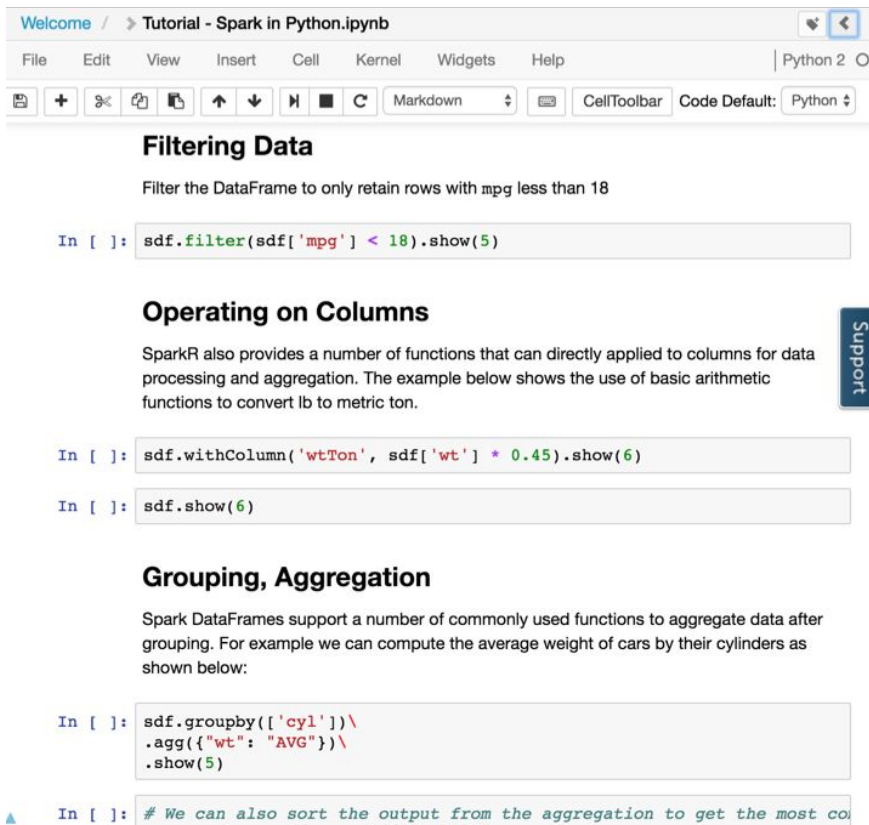


Play around with free data sets



- Government Data - [Data.gov](https://data.gov)
- World Health Organisation: <http://apps.who.int/gho/data/node.home>
- UCI Machine Learning Repository: <https://archive.ics.uci.edu/ml/datasets.html>
- Data World - [Data.world](https://data.world)

Document your learning with Notebook



The screenshot shows a Jupyter Notebook interface with the title 'Tutorial - Spark in Python.ipynb'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, undo, redo, and other actions. The notebook content is divided into three sections:

- Filtering Data**
Filter the DataFrame to only retain rows with mpg less than 18

```
In [ ]: sdf.filter(sdf['mpg'] < 18).show(5)
```
- Operating on Columns**
SparkR also provides a number of functions that can directly applied to columns for data processing and aggregation. The example below shows the use of basic arithmetic functions to convert lb to metric ton.

```
In [ ]: sdf.withColumn('wtTon', sdf['wt'] * 0.45).show(6)
```

```
In [ ]: sdf.show(6)
```
- Grouping, Aggregation**
Spark DataFrames support a number of commonly used functions to aggregate data after grouping. For example we can compute the average weight of cars by their cylinders as shown below:

```
In [ ]: sdf.groupby(['cyl'])\
        .agg({'wt': 'AVG'})\
        .show(5)
```

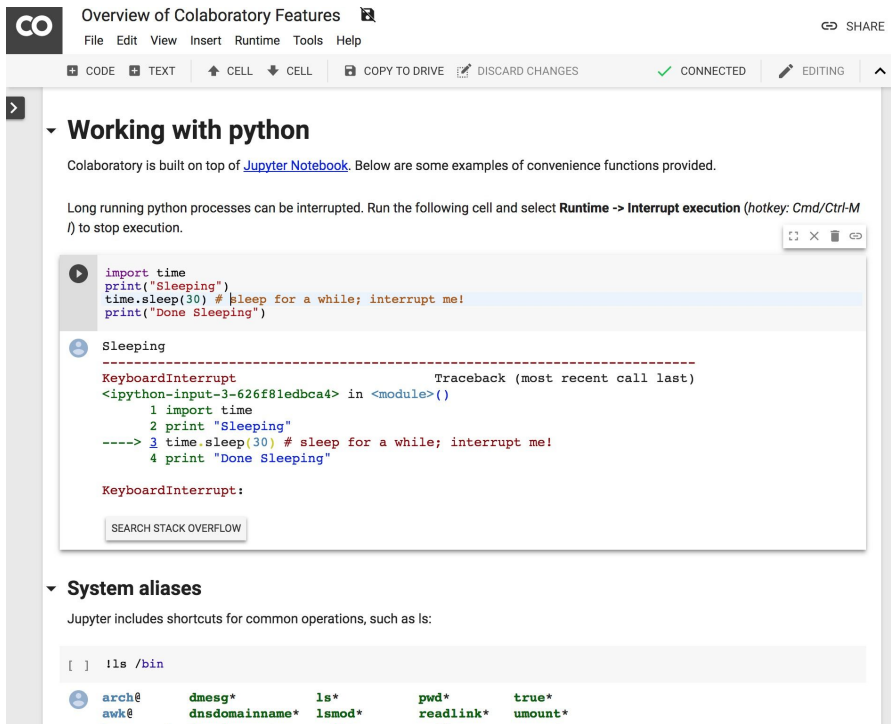
```
In [ ]: # We can also sort the output from the aggregation to get the most co
```

Jupyter Notebook


<https://datascientistworkbench.com/>

- Free
- Same as “Kernel” on Kaggle
- Offline (with Anaconda) or Online
- Support Python with Anaconda installation
- <https://anaconda.org/>
- Support R with IRKernel

Document your learning with Notebook



The screenshot shows the Google Colaboratory web interface. At the top, there's a navigation bar with the Colab logo, 'Overview of Colaboratory Features', and a 'SHARE' button. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. A secondary bar contains icons for 'CODE', 'TEXT', 'CELL', 'COPY TO DRIVE', 'DISCARD CHANGES', 'CONNECTED', and 'EDITING'. The main content area is titled 'Working with python' and contains text explaining that Colaboratory is built on top of Jupyter Notebook. It provides an example of a Python cell that prints 'Sleeping', sleeps for 30 seconds, and prints 'Done Sleeping'. The cell is interrupted, and the output shows a 'KeyboardInterrupt' traceback. Below this, there's a section titled 'System aliases' which lists common shortcuts like 'ls', 'pwd', 'true', etc.

Overview of Colaboratory Features  [SHARE](#)

File Edit View Insert Runtime Tools Help

CODE TEXT CELL COPY TO DRIVE DISCARD CHANGES CONNECTED EDITING

Working with python

Colaboratory is built on top of [Jupyter Notebook](#). Below are some examples of convenience functions provided.

Long running python processes can be interrupted. Run the following cell and select **Runtime** -> **Interrupt execution** (hotkey: **Cmd/Ctrl-M**) to stop execution.

```
import time
print("Sleeping")
time.sleep(30) # sleep for a while; interrupt me!
print("Done Sleeping")
```

Sleeping

```
KeyboardInterrupt                               Traceback (most recent call last)
<ipython-input-3-626f81edbc4> in <module>()
      1 import time
      2 print "Sleeping"
----> 3 time.sleep(30) # sleep for a while; interrupt me!
      4 print "Done Sleeping"

KeyboardInterrupt:
```

[SEARCH STACK OVERFLOW](#)

System aliases

Jupyter includes shortcuts for common operations, such as ls:


```
[ ] !ls /bin
```

alias	command
arch	!arch
awk	!awk
dmesg	!dmesg
dnsdomainname	!dnsdomainname
ls	!ls
lsmmod	!lsmmod
pwd	!pwd
readlink	!readlink
true	!true
umount	!umount

Google Colaboratory

<https://colab.research.google.com/notebook>

- Free
- Online
- Support Python 2 and 3
- Shareable link
- Collaboration with people (like Google Docs)



Congratulations!
You have become **Kagglers!**

RECAP

- Kaggle is a good way to practice machine learning, data analysis skills
- Kaggle's goal: Turn data into prediction results
- Data dictionary is important for understanding the data set
- Find the target variable first

RECAP: Building Model

How to build a prediction model

1. Prepare features and target variables
2. Use train data to train the model
3. Feed test data into the model to get prediction results
4. Optimize the model
5. Submit to [Kaggle](#)

