

Specialist Programme on Artificial Intelligence for IT & ITES Industry

Intelligent Process Automation (IPA) Part 2/2

By GU Zhan (Sam)

Zhan.Gu@nus.edu.sg

Singapore e-Government Leadership Centre
National University of Singapore

© 2020 NUS. The contents contained in this document may not be reproduced in any form or by any means,
without the written permission of ISS, NUS other than for the purpose for which it has been supplied.



The logo features a red staircase with three steps. The first step contains the word "Inspire" in white. The second step contains the word "Lead" in white. The third step contains the word "Transform" in white. The steps are set against a white background.

Agenda

1.1 Intelligent Agent For IPA

1.2 Intelligent Agent Using Cloud AI

1.3 Intelligent Agent System Show Case

1.4 Intelligent Agent Workshop: Virtual Assistant

1.1 INTELLIGENT AGENT FOR IPA

What's intelligence of an agent?

“Intelligence measures an agent’s ability to achieve goals in a wide range of environments.”

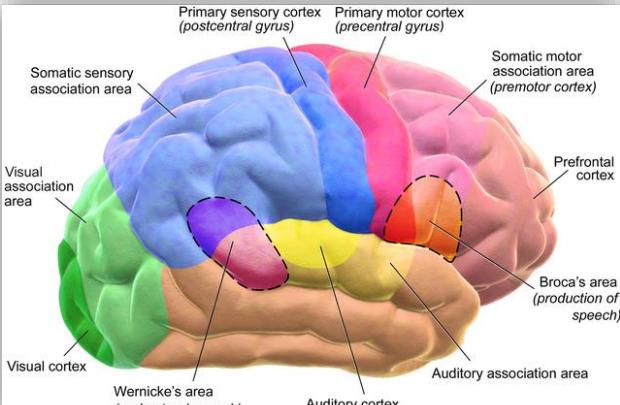
A Universal Measure of Intelligence for Artificial Agents, Legg & Hutter 2007

$$\Upsilon(\pi) := \sum_{\mu \in E} 2^{-K(\mu)} V_\mu^\pi$$

Sum over environments

Measure of Intelligence
Complexity penalty
Value achieved

1.1 INTELLIGENT AGENT FOR IPA



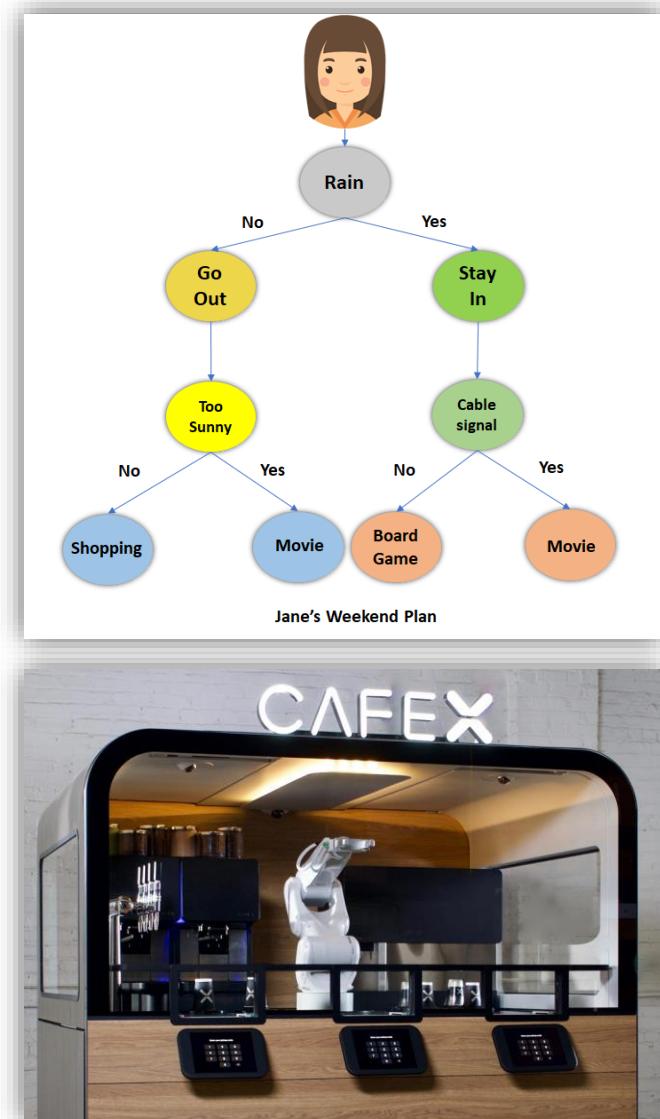
Thinking vs Acting (acting = behaviour)
Human vs Rational (rationality = doing the right thing)

Systems that think like humans
(cognitive science)

Systems that act like humans
(c.f. Turing test)

Systems that think rationally
(logic/laws of thought)

Systems that act rationally
(agents)



Source <https://slideplayer.com/slide/4644026/15/images/20/Systems+that+think+like+humans+%28cognitive+science%29.jpg>

1.1 INTELLIGENT AGENT FOR IPA

Agents & AI: Aren't agents just the AI or intelligent system project?

- **AI aims to build systems that can (ultimately) understand human language, recognize & understand scenes, use common sense, etc. – all of which are very hard**
- **When building an agent, we simply want a system that can choose the right action to perform, typically in a very limited domain**

“Intelligent agents are ninety-nine percent computer science and one percent AI”

-- Oren Etzioni

1.1 INTELLIGENT AGENT FOR IPA

Classes

[Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, chpt. 2]

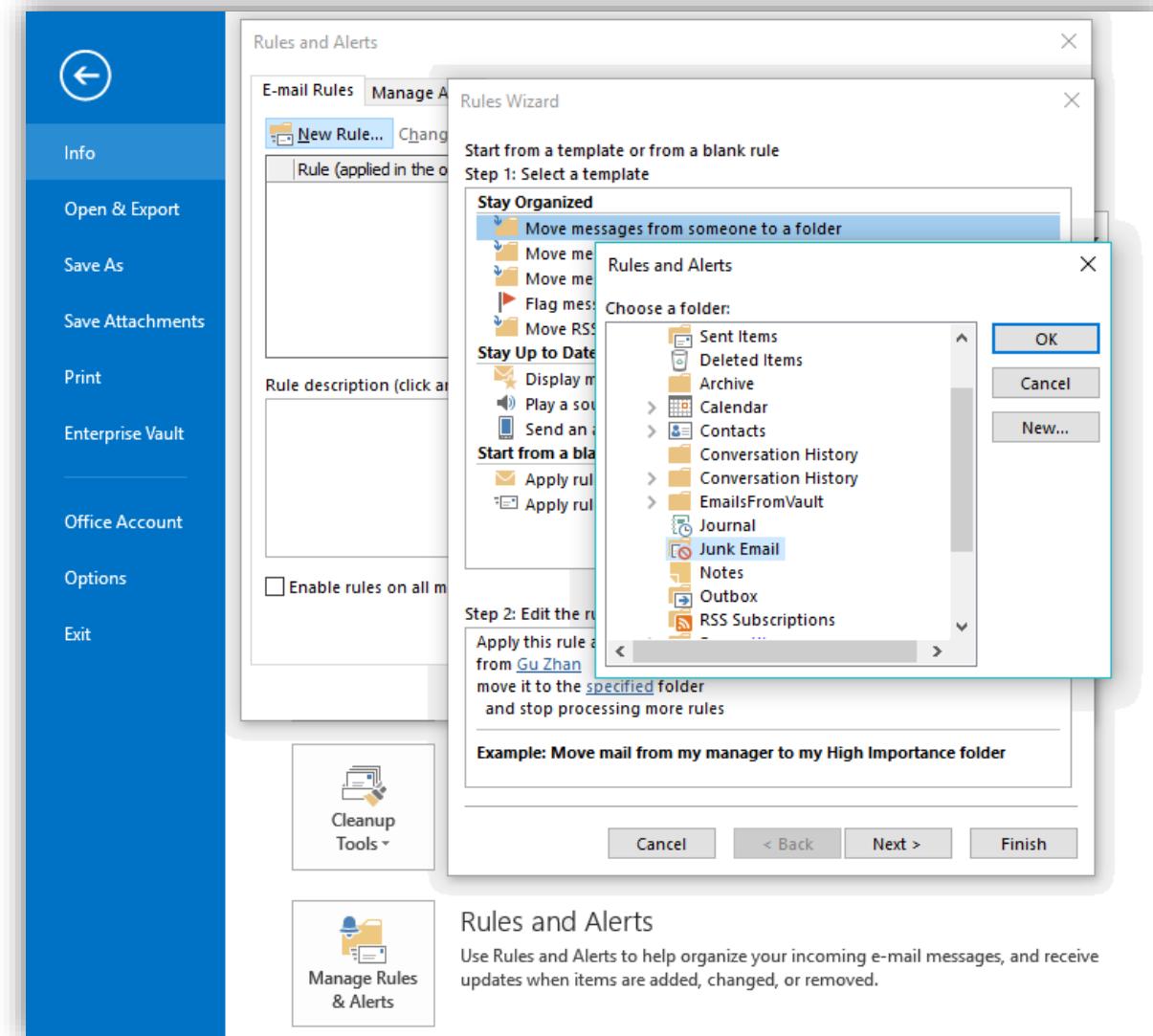
"The main unifying theme (in AI) is the idea of an intelligent agent. We define AI as the study of agents that receive **percepts** from the environment and perform **actions**. Each such agent implements **a function that maps percept sequences to actions**, and we cover different ways to represent these functions, such as reactive agents, real-time planners, and decision-theoretic systems. We explain the role of learning as extending the reach of the designer into unknown environments..."

- **Simple reflex agents**
- **Model-based reflex agents**
- **Goal-based agents**
- **Utility-based agents**
- **Learning agents**

1.1 INTELLIGENT AGENT FOR IPA

Applications of agent class:

- **Simple reflex agents** : Simple reflex agents act only on the basis of the current percept, ignoring the rest of the percept history. The agent function is based on the **condition-action rule**: "if condition, then action".
- E.g. Email Routing Agent



1.1 INTELLIGENT AGENT FOR IPA

Applications of agent class:

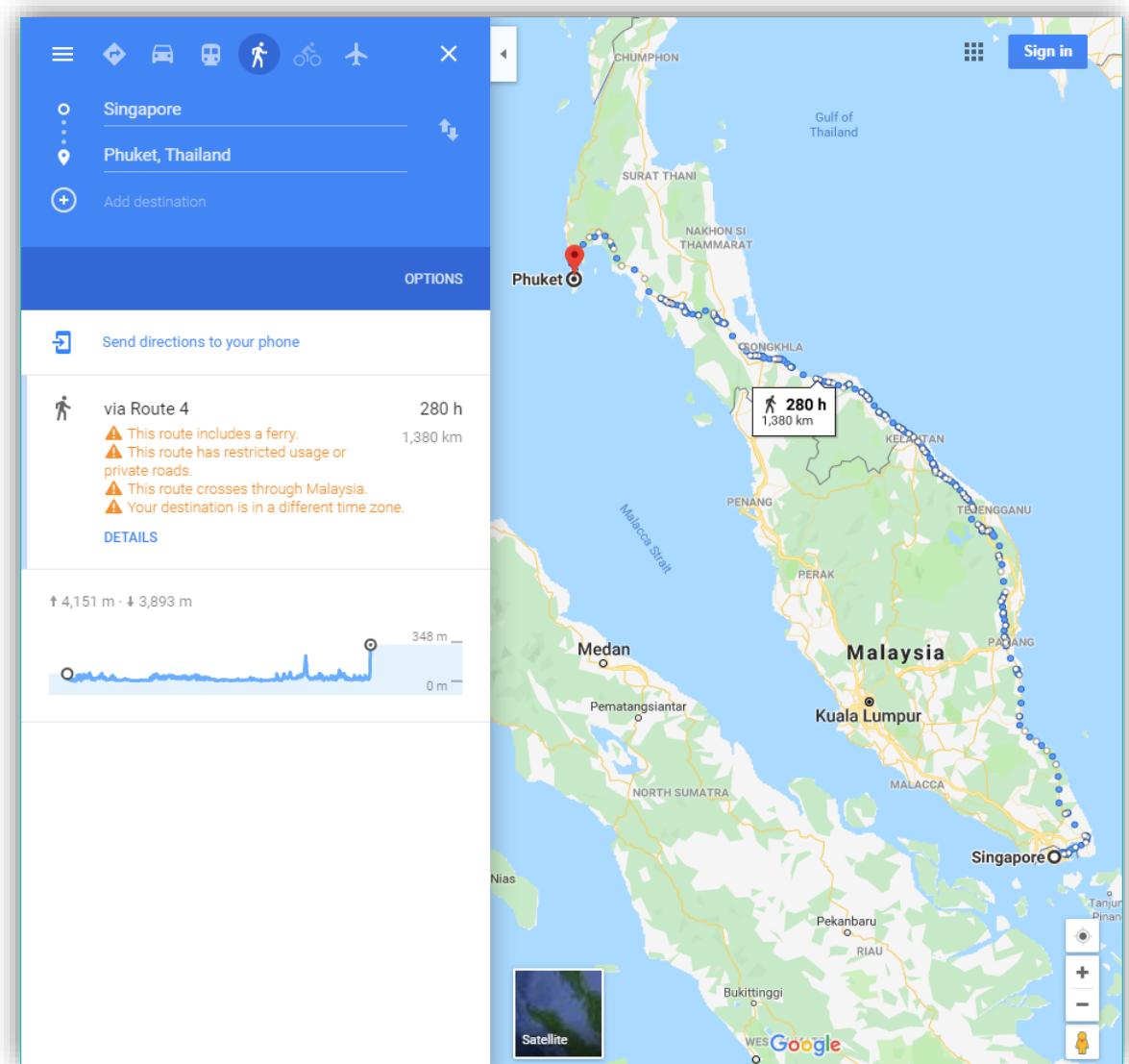
- **Model-based reflex agents** : A model-based agent can handle partially observable environments. Its current state is stored inside the agent maintaining some kind of structure which describes the part of the world which cannot be seen. This knowledge about "how the world works" is called a **model** of the world, hence the name "model-based agent".
- E.g. Finite State (Copy) Machine; Knowledge based (agent) system



1.1 INTELLIGENT AGENT FOR IPA

Applications of agent class:

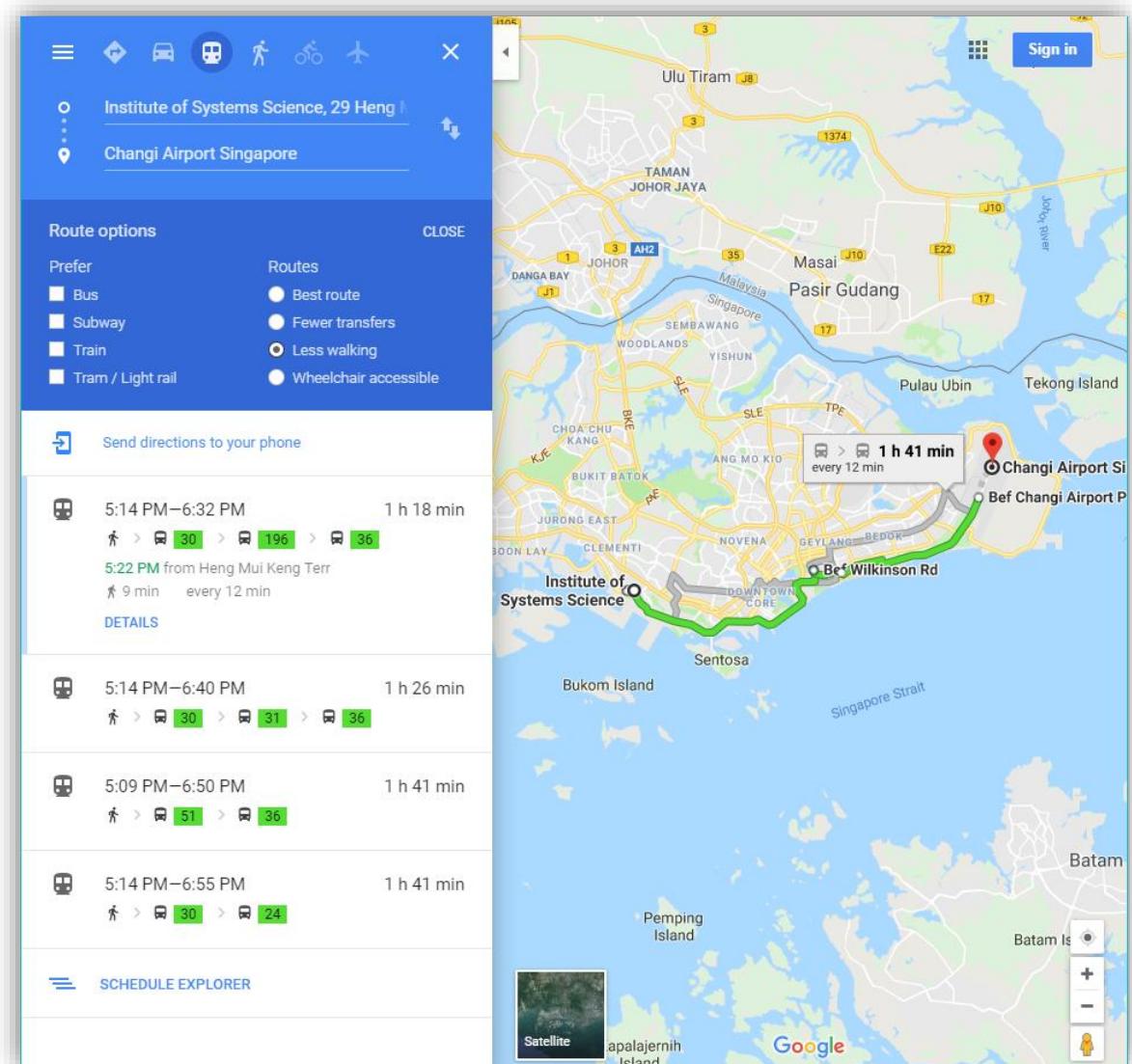
- **Goal-based agents** : Goal-based agents further expand on the capabilities of the model-based agents, by using "goal" information. Goal information describes situations that are desirable. This allows the agent a way to choose among **multiple possibilities**, selecting the one which reaches a **goal state**. Search and planning are the subfields of artificial intelligence devoted to finding action sequences that achieve the agent's goals.
- E.g. Route Finder (by walk/ferry)



1.1 INTELLIGENT AGENT FOR IPA

Applications of agent class:

- **Utility-based agents** : Goal-based agents only distinguish between goal states and non-goal states. It is possible to define a measure of how desirable a particular state is. This measure can be obtained through the use of a **utility function** which maps a state to a measure/fitness of the utility of the state. A more general performance measure should allow a comparison of different interim solutions/states according to exactly how happy they would make the agent. The term utility can be used to describe how "happy" the agent is.
- E.g. Best Route Finder (less walking)



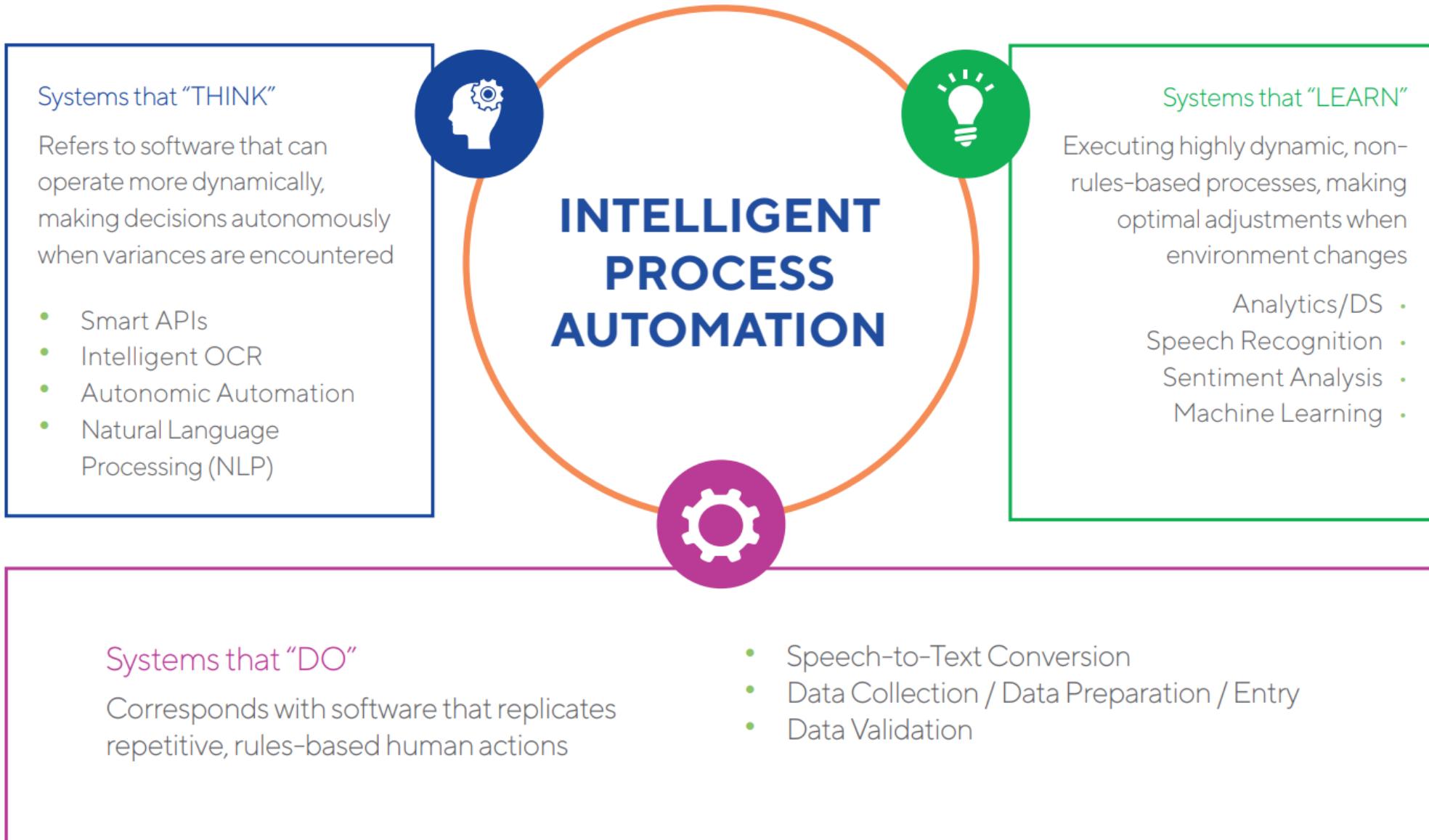
1.1 INTELLIGENT AGENT FOR IPA

Question:

What's the agent class of pip?

```
(iss-env-py3) iss-user@iss-vm:~$ pip install face_recognition
Collecting face_recognition
  Using cached https://files.pythonhosted.org/packages/3f/ed/ad9a28042f373d4633fc8b49109b623597d6f193d3bbbef7780a5ee8eef2/face_recognition-1.2.3-py2.py3-none-any.whl
Requirement already satisfied: dlib>=19.7 in ./anaconda3/envs/iss-env-py3/lib/python3.6/site-packages (from face_recognition) (19.16.0)
Requirement already satisfied: face-recognition-models>=0.3.0 in ./anaconda3/envs/iss-env-py3/lib/python3.6/site-packages (from face_recognition) (0.3.0)
Requirement already satisfied: Click>=6.0 in ./anaconda3/envs/iss-env-py3/lib/python3.6/site-packages (from face_recognition) (6.7)
Requirement already satisfied: Pillow in ./anaconda3/envs/iss-env-py3/lib/python3.6/site-packages (from face_recognition) (5.0.0)
Requirement already satisfied: numpy in ./anaconda3/envs/iss-env-py3/lib/python3.6/site-packages (from face_recognition) (1.14.0)
Installing collected packages: face-recognition
Successfully installed face-recognition-1.2.3
You are using pip version 18.1, however version 19.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(iss-env-py3) iss-user@iss-vm:~$
```

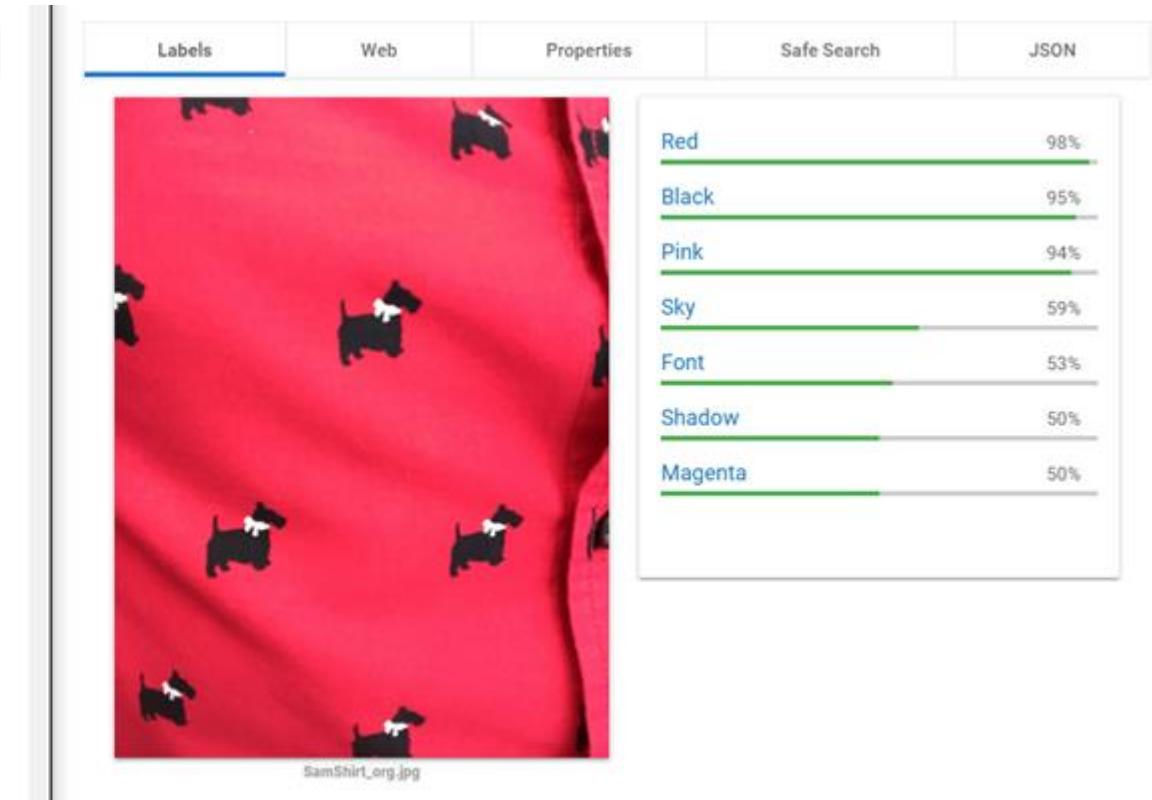
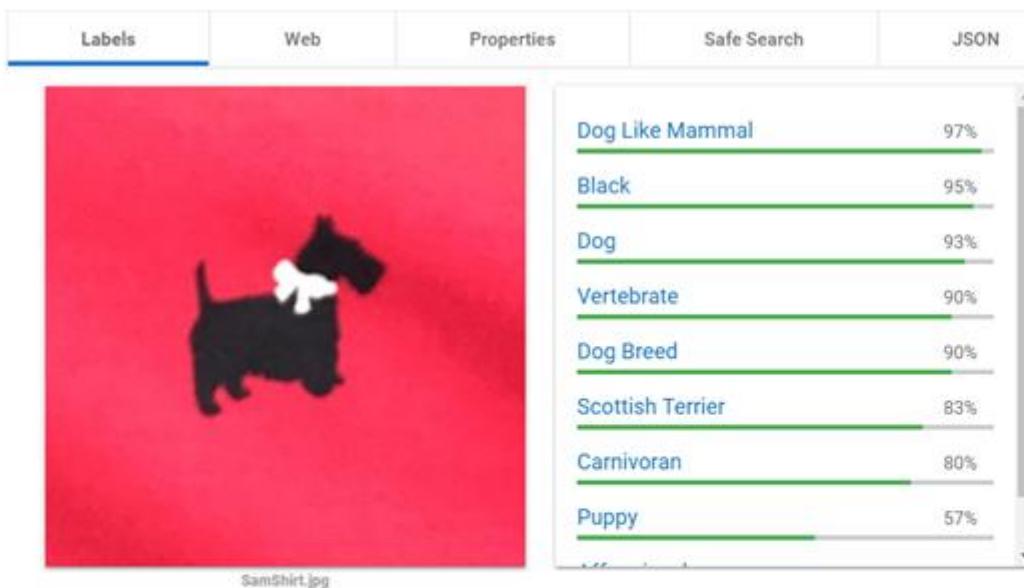
1.1 INTELLIGENT AGENT FOR IPA



1.2 INTELLIGENT AGENT USING CLOUD AI

1.2 INTELLIGENT AGENT USING CLOUD AI

Visual Processing



[Source https://cloud.google.com/vision/](https://cloud.google.com/vision/)

1.2 INTELLIGENT AGENT USING CLOUD AI

Sentiment Analysis

The screenshot shows the Google Cloud AI & Machine Learning Products interface. Under the 'Sentiment' tab, it displays the 'Document & Sentence Level Sentiment' section. An example sentence is analyzed: "One of the most useful things we learnt at ISS was to be analytical and to be able to adapt, especially in an industry that is changing so quickly. What ISS gave us was an invaluable reference point." The overall sentiment score is 0.9 with a magnitude of 1.9. Below this, a color-coded scale shows 'Score Range' from -1.0 to 1.0, with segments for -1.0 -- -0.25 (red), -0.25 -- 0.25 (yellow), and 0.25 -- 1.0 (green).

Language Processing

The screenshot shows the Google Cloud AI & Machine Learning Products interface. Under the 'Entities' tab, it displays the 'Entity Level Sentiment' section. It lists four entities with their respective sentiment scores, magnitudes, and categories:

Entity	Category	Sentiment	Magnitude
1. One	OTHER	Score: 0.9	Magnitude: 0.9
2. things	OTHER	Score: 0.9	Magnitude: 0.9
3. ISS	ORGANIZATION	Score: 0	Magnitude: 0.1
4. industry	OTHER	Score: 0.3	Magnitude: 0.7

Source <https://cloud.google.com/natural-language/>

1.2 INTELLIGENT AGENT USING CLOUD AI

Speech Recognition



Google Cloud Why Google Products Solutions Pricing Security Documentation C > Search Console Sign in

Contact sales Try free

AI & Machine Learning Products

Input type: Microphone File upload

Language: English (United States)

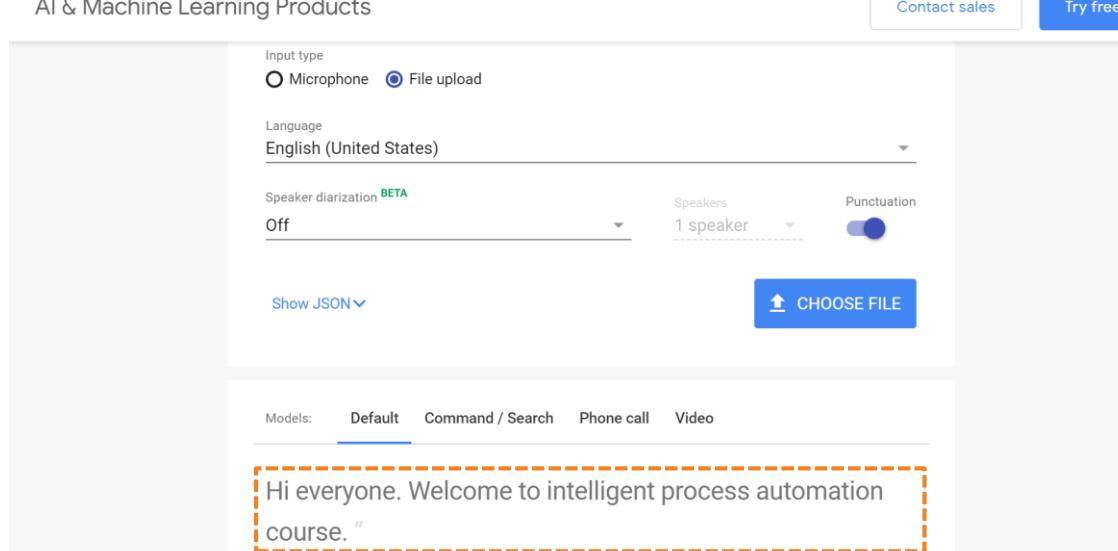
Speaker diarization BETA: Off

Speakers: 1 speaker Punctuation:

Show JSON ▾ CHOOSE FILE

Models: Default Command / Search Phone call Video

Text to speak: Hi everyone. Welcome to intelligent process automation course.



[Source](https://cloud.google.com/speech-to-text/) https://cloud.google.com/speech-to-text/

Speech Synthesis

"Hi everyone! It's great to meet you in intelligent process automation course!"

Google Cloud Why Google Products Solutions Pricing Security Documentation > Search Console Sign in

Contact sales Try free

AI & Machine Learning Products

Type what you want, select a language then click "Speak It" to hear.

Text to speak: Hi everyone! It's great to meet you in intelligent process automation course! text ssml

Language / locale: English (United States)

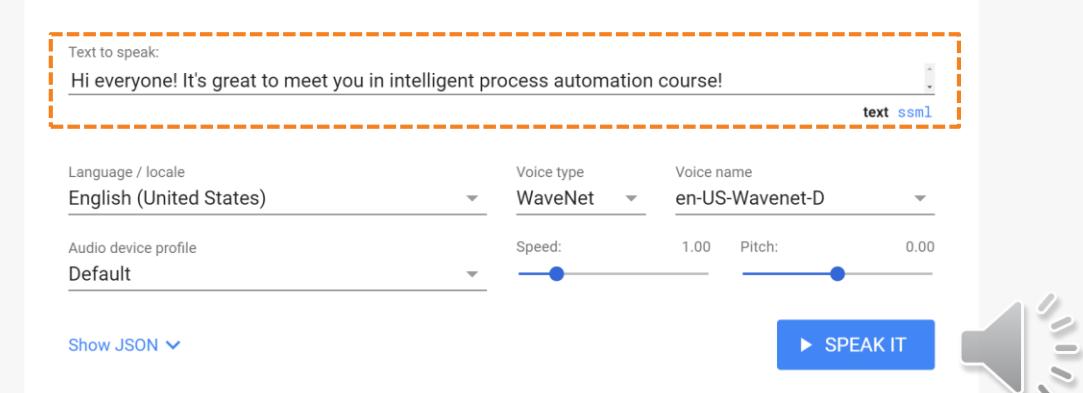
Voice type: WaveNet

Voice name: en-US-Wavenet-D

Audio device profile: Default

Speed: 1.00 Pitch: 0.00

Show JSON ▾ SPEAK IT



[Source](https://cloud.google.com/text-to-speech/) https://cloud.google.com/text-to-speech/

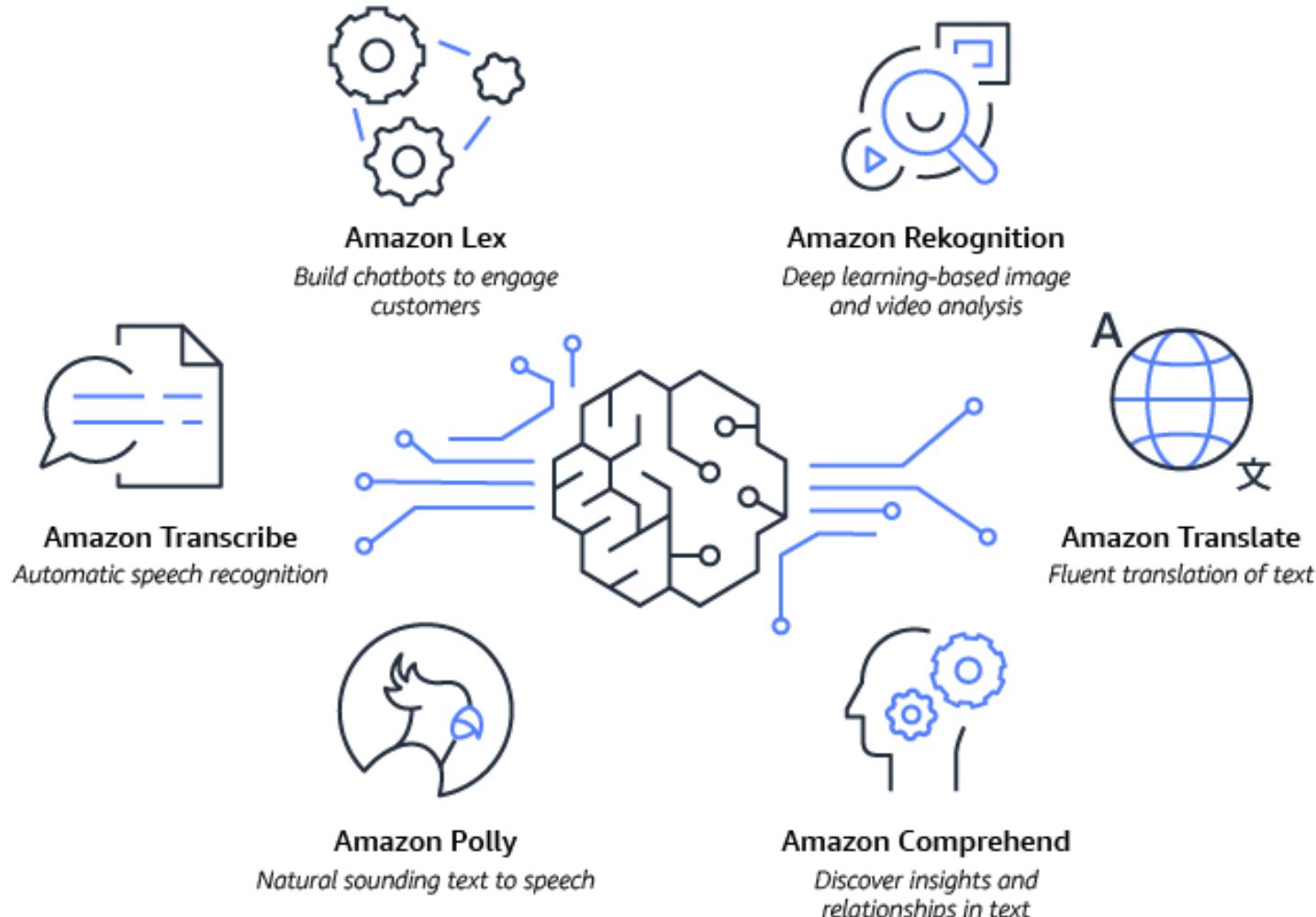
1.2 INTELLIGENT AGENT USING CLOUD AI

- **Artificial Intelligence functional modules**
 - Cloud AI services
 - Cloud AI IPA use case

1.2 INTELLIGENT AGENT USING CLOUD AI

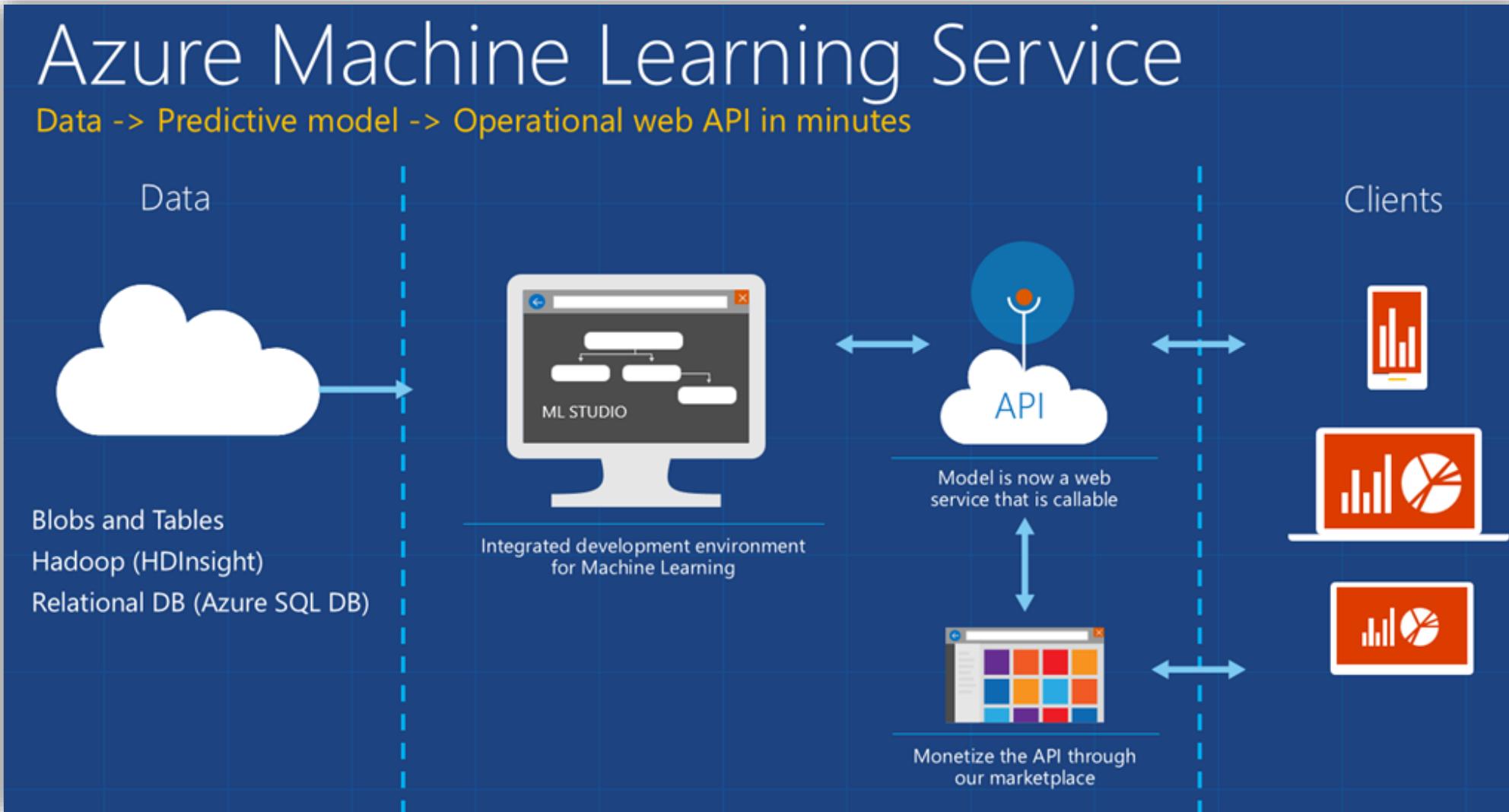
- Artificial Intelligence functional modules
 - Cloud AI services
 - Cloud AI IPA use case

1.2 INTELLIGENT AGENT USING CLOUD AI : SERVICES



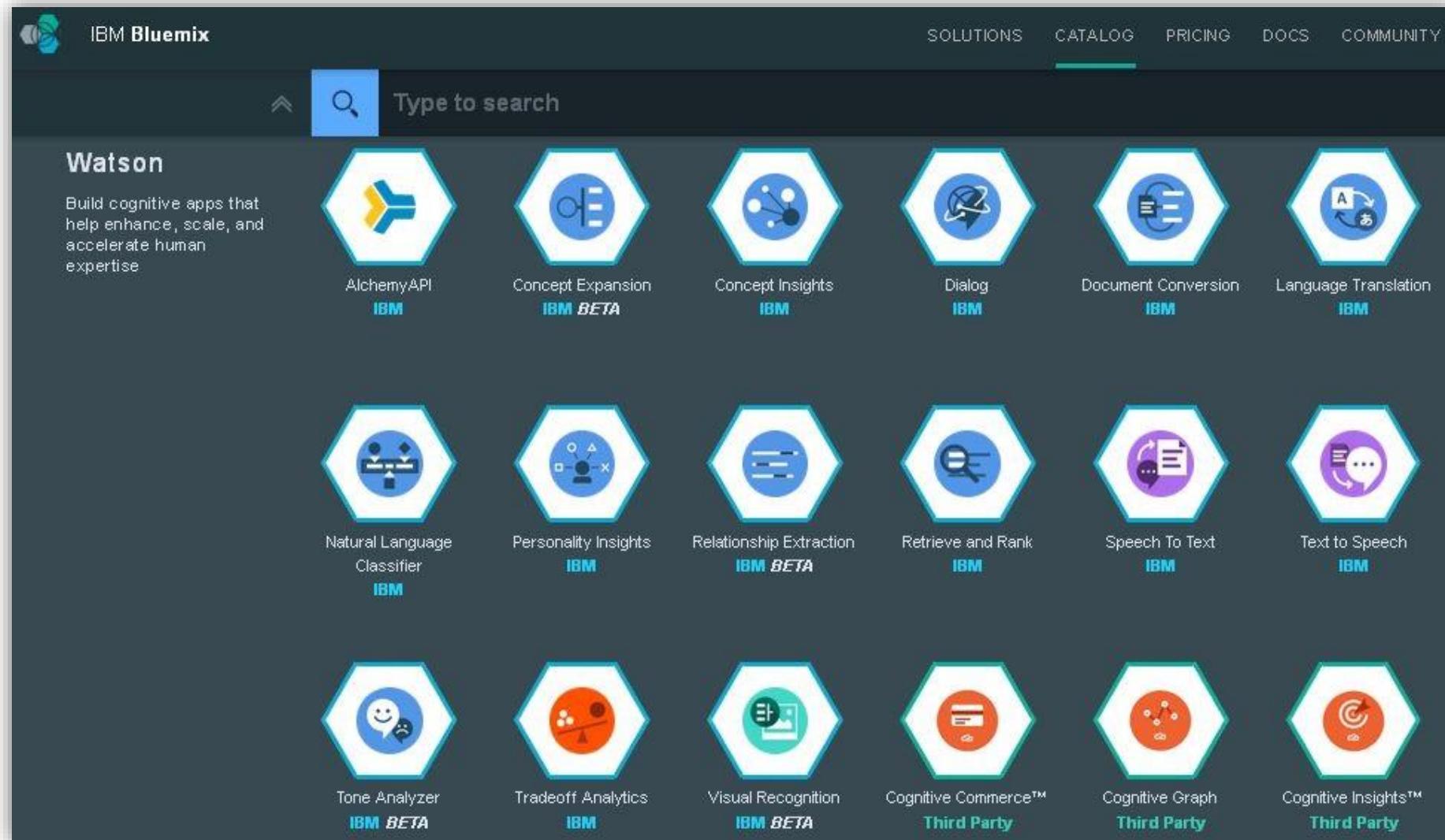
Source <https://aws.amazon.com/blogs/machine-learning/announcing-the-artificial-intelligence-ai-hackathon-build-intelligent-applications-using-machine-learning-apis-and-serverless>

1.2 INTELLIGENT AGENT USING CLOUD AI : SERVICES



Source <https://msdnshared.blob.core.windows.net/media/2017/03/image423.png>

1.2 INTELLIGENT AGENT USING CLOUD AI : SERVICES



Source <https://developer.qualcomm.com/sites/default/files/attachments/bluemixcatalogwatson.jpg>

1.2 INTELLIGENT AGENT USING CLOUD AI : SERVICES

Baidu brain, open the world's leading AI service

Voice technology Image technology Face and body recognition Video technology

AR and VR Natural language processing Data intelligence Knowledge map

Language processing basic technology

- lexical analysis
- Word vector representation
- Word similarity
- Dependency parsing analysis
- DNN language model
- Short text similarity 热门

Text review 新品

machine translation

- Universal translation API
- Customized translation API
- Voice translation SDK 新品
- Photo translation SDK 热门

[Source](https://ai.baidu.com/) https://ai.baidu.com/

1.2 INTELLIGENT AGENT USING CLOUD AI : SERVICES

The screenshot shows the Alibaba Cloud website with a dark theme. At the top, there's a navigation bar with links for Contact Sales, Search, International English, Cart, Console, and Log In. Below the navigation is a main menu with Why Us, Products, Solutions, Pricing, Marketplace, Resources, Support, Documentation, and a prominent blue 'Free Account' button. The main content area features a large heading 'Machine Learning Platform For AI' and a subtext: 'An end-to-end platform that provides various machine learning algorithms to meet your data mining and analysis requirements.' Below this are three buttons: 'Buy Now' (blue), 'Console' (white), and 'Contact Sales' (white). To the right is a 3D wireframe cube diagram with a central blue cube. At the bottom, there are tabs for Overview (selected), Pricing, Documentation, and a 'Contact Us' button. A sidebar on the left contains a link to a Data Sheet. The URL https://common-buy-intl.aliyun.com/?commodityCode=pa1_intl#/buy is visible at the bottom.

Source <https://www.alibabacloud.com/product/machine-learning>

1.2 INTELLIGENT AGENT USING CLOUD AI : SERVICES

The screenshot shows the Huawei ModelArts product page. At the top, there's a navigation bar with the Huawei logo, search bar, and links for China Station, Console, Login, and Register. The main heading is "ModelArts". Below it, a brief description states: "ModelArts is a one-stop development platform for AI developers. With data preprocessing, semi-automated data labeling, distributed training, automated model building, and model deployment on the device, edge, and cloud, ModelArts helps AI developers build models...". It mentions pay-per-use billing starting at ¥0.8/h. There are buttons for "Try Now", "Pricing Details", "Get Discounts", "Get Started", and "Contact Us". A large central image shows a smartphone displaying a neural network diagram. Below this, there are three sections: "Sound Classification" (NEW), "New Data Labeling" (NEW), and "Audio and Text Labeling" (NEW). The "Product Advantages" section lists four items: "One-Stop Platform" (represented by a document icon), "Easy to Use" (represented by a gear icon), "Excellent Performance" (represented by a graph icon), and "High Flexibility" (represented by a checkmark icon). A "Contact Us" button is located on the right side of the advantages section.

[Source](https://www.huaweicloud.com/en-us/product/modelarts.html) https://www.huaweicloud.com/en-us/product/modelarts.html

1.2 INTELLIGENT AGENT USING CLOUD AI : SERVICES

Machine Learning engine and APIs

Pre-trained ML models



Vision API



Speech API



Jobs API

Custom ML models



TensorFlow



Machine Learning Engine



Translation API



Natural Language API

Source <https://image.slidesharecdn.com/cloud-bkk-june-2017-170624082709/95/data-science-on-google-cloud-platform-44-638.jpg>

1.2 INTELLIGENT AGENT USING CLOUD AI

- **Artificial Intelligence functional modules**
 - Cloud AI services
 - **Cloud AI IPA use case**

1.2 INTELLIGENT AGENT USING CLOUD AI : IPA USE CASE 1

Analyse face emotions

- Virtual worker obtains learner's profile image shown on ISS website.
- Virtual worker calls cloud AI API to conduct face detection and emotion analysis.

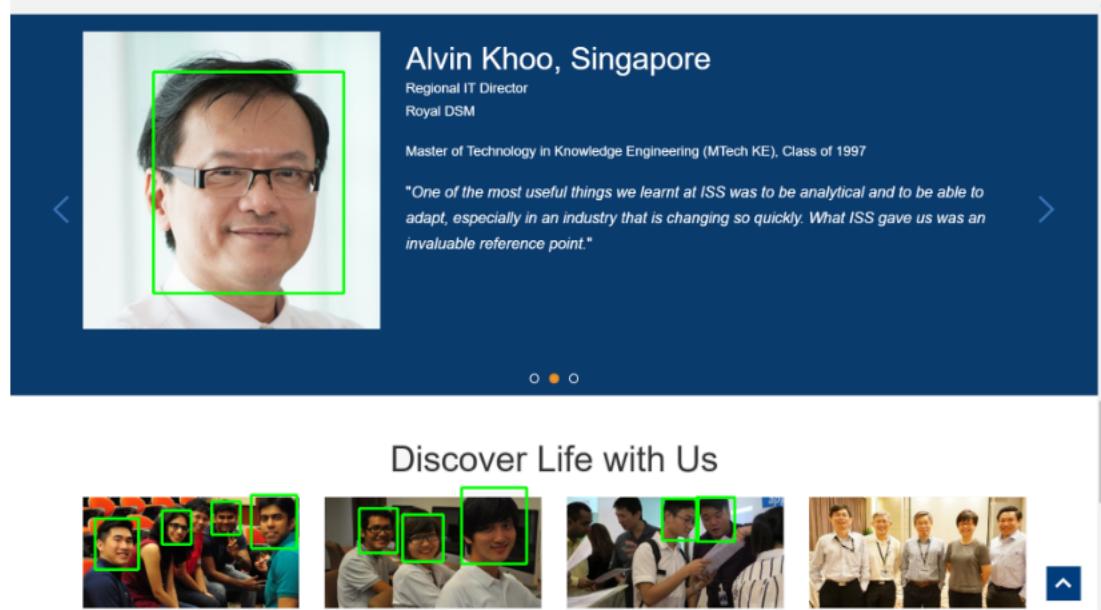
```
1 https://ca.yahoo.com
2 // automation flow files usually start with an URL to tell TagUI where to go
3 // files can also start with // for comments, or no URL if it's not web-related
4
5 // this flow searches github on Yahoo and captures screenshot of results
6 // for issues or questions, kindly feedback on GitHub or ksch@aisingapore.org
7 // see cheatsheet for steps, conditions, finding element identifiers, etc
8 // https://github.com/kelaberetiv/TagUI#cheat-sheet
9
10 echo 'NOTE - this sample may not work for EU users due to website changes for GDPR'
11 echo 'EU users may have to modify script to first click on announcement OK button'
12 echo ''
13 // lines which begin with // means user comments, which are ignored during execution
14
15 // use live step to try steps or code interactively for Chrome / visual automation
16 // this helps accelerate automation script development as you can test each step
17 // live
```

Source

https://github.com/telescopeteuser/workshop_blog/blob/master/wechat_tool_py3_local/lesson_2_py3_local.ipynb

```
In [22]: didi_face_detection('image//ISS-testimony-sentiment.png', 'image//ISS-testimony-sentimentDetectedFace.png')
)
Found 10 faces
Writing to file image//ISS-testimony-sentimentDetectedFace.png
```

```
In [23]: IPython.display.Image(filename='image//ISS-testimony-sentimentDetectedFace.png')
```



Face Detection : Identify sentiment and emotion from human face

<https://cloud.google.com/vision/docs/detecting-faces>

```
In [24]: def didi_face_detection_emotion(image):
    """Detects faces in an image."""
    from google.cloud import vision

#####
#     client = vision.ImageAnnotatorClient()
#
#     client = vision.ImageAnnotatorClient.from_service_account_json(
#         "/media/sf_vm_shared_folder/000-cloud-api-key/mtech-ai-7b7e049cf5f6.json")
#####

    with io.open(image, 'rb') as image_file:
        content = image_file.read()

    image = vision.types.Image(content=content)

    response = client.face_detection(image=image)
    faces = response.face_annotations

    # Names of Likelihood from google.cloud.vision.enums
    likelihood_name = ('UNKNOWN', 'VERY_UNLIKELY', 'UNLIKELY', 'POSSIBLE',
                        'LIKELY', 'VERY_LIKELY')
```

```
# Names of likelihood from google.cloud.vision.enums
likelihood_name = ('UNKNOWN', 'VERY_UNLIKELY', 'UNLIKELY', 'POSSIBLE',
                    'LIKELY', 'VERY_LIKELY')
print('Found {} face{} : Emotion {}'.format(len(faces), '' if len(faces) == 1 else 's'))

for face in faces:
    print(' * anger      : {}'.format(likelihood_name[face.anger_likelihood]))
    print(' * joy       : {}'.format(likelihood_name[face.joy_likelihood]))
    print(' * sorrow    : {}'.format(likelihood_name[face.sorrow_likelihood]))
    print(' * surprise   : {}'.format(likelihood_name[face.surprise_likelihood]))

    vertices = ('[{}{},{}{}]''.format(vertex.x, vertex.y)
                for vertex in face.bounding_poly.vertices)

    print(' * face bounds : {}'\n''.format(','.join(vertices)))
```

```
In [26]: didi_face_detection_emotion('image//ISS-testimony-sentiment.png')
```

```
Found 10 faces : Emotion
* anger      : VERY_UNLIKELY
* joy       : VERY_LIKELY
* sorrow    : VERY_UNLIKELY
* surprise   : VERY_UNLIKELY
* face bounds : (250,125),(583,125),(583,512),(250,512)

* anger      : VERY_UNLIKELY
* joy       : VERY_UNLIKELY
* sorrow    : VERY_UNLIKELY
* surprise   : VERY_UNLIKELY
* face bounds : (421,865),(501,865),(501,958),(421,958)

* anger      : VERY_UNLIKELY
* joy       : VERY_LIKELY
* sorrow    : VERY_UNLIKELY
* surprise   : VERY_UNLIKELY
* face bounds : (148,906),(224,906),(224,995),(148,995)

* anger      : VERY_UNLIKELY
* joy       : VERY_LIKELY
* sorrow    : VERY_UNLIKELY
* surprise   : VERY_UNLIKELY
* face bounds : (789,852),(903,852),(903,985),(789,985)

* anger      : VERY_UNLIKELY
* joy       : VERY_LIKELY
* sorrow    : VERY_UNLIKELY
* surprise   : VERY_UNLIKELY
* face bounds : (610,888),(677,888),(677,966),(610,966)

* anger      : VERY_UNLIKELY
* joy       : VERY_LIKELY
* sorrow    : VERY_UNLIKELY
* surprise   : VERY_UNLIKELY
* face bounds : (686,899),(757,899),(757,981),(686,981)

* anger      : VERY_UNLIKELY
* joy       : VERY_UNLIKELY
* sorrow    : VERY_UNLIKELY
* surprise   : VERY_UNLIKELY
* face bounds : (1200,869),(1267,869),(1267,946),(1200,946)

* anger      : VERY_UNLIKELY
* joy       : LIKELY
```

1.2 INTELLIGENT AGENT USING CLOUD AI : IPA USE CASE 2

Analyse testimony sentiment from website

- Virtual worker obtains learner's textual testimony shown on ISS website.
- Virtual worker calls cloud AI API to analyse sentiment.

```
1 https://ca.yahoo.com
2 // automation flow files usually start with an URL to tell TagUI where to go
3 // files can also start with // for comments, or no URL if it's not web-related
4
5 // this flow searches github on Yahoo and captures screenshot of results
6 // for issues or questions, kindly feedback on GitHub or ksch@aisingapore.org
7 // see cheatsheet for steps, conditions, finding element identifiers, etc
8 // https://github.com/kelaberetiv/TagUI#cheat-sheet
9
10 echo 'NOTE - this sample may not work for EU users due to website changes for GDPR'
11 echo 'EU users may have to modify script to first click on announcement OK button'
12 echo ''
13
14 // lines which begin with // means user comments, which are ignored during execution
15
16 // use live step to try steps or code interactively for Chrome / visual automation
17 // this helps accelerate automation script development as you can test each step
18 // live
```

Source

https://github.com/telescopeteuser/workshop_blog/blob/master/wechat_tool_py3_local/lesson_4_py3_local.ipynb



Alvin Khoo, Singapore
Regional IT Director
Royal DSM

Master of Technology in Knowledge Engineering (MTech KE), Class of 1997

"One of the most useful things we learnt at ISS was to be analytical and to be able to adapt, especially in an industry that is changing so quickly. What ISS gave us was an invaluable reference point."

Discover Life with Us



```
In [4]: # Imports the Google Cloud client library
from google.cloud import language
from google.cloud.language import enums
from google.cloud.language import types

# (1) Instantiates a client - using GOOGLE_APPLICATION_CREDENTIALS
# client = language.LanguageServiceClient()

# (2) Instantiates a client - using 'service account json' file
client = language.LanguageServiceClient.from_service_account_json(
    '/media/sf_vm_shared_folder/000-cloud-api-key/mtech-ai-7b7e049cf5f6.json')

# The text to analyze
text=u'Alvin Khoo: One of the most useful things we learnt at ISS was to be analytical and to be able to adapt, \
especially in an industry that is changing so quickly. What ISS gave us was an invaluable reference point.'

document = types.Document(
    content=text,
    type=enums.Document.Type.PLAIN_TEXT)

# Detects the sentiment of the text
sentiment = client.analyze_sentiment(document=document).document_sentiment

print('Text: {}'.format(text))
print(' ')
print('Sentiment: {}, {}'.format(sentiment.score, sentiment.magnitude))

Text: Alvin Khoo: One of the most useful things we learnt at ISS was to be analytical and to be able to adapt, \
especially in an industry that is changing so quickly. What ISS gave us was an invaluable reference point.

Sentiment: 0.8999999761581421, 1.899999976158142
```

- `documentSentiment` contains the overall sentiment of the document, which consists of the following fields:
 - `score` of the sentiment ranges between `-1.0` (negative) and `1.0` (positive) and corresponds to the overall emotional leaning of the text.
 - `magnitude` indicates the overall strength of emotion (both positive and negative) within the given text, between `0.0` and `+inf`. Unlike `score`, `magnitude` is not normalized; each expression of emotion within the text (both positive and negative) contributes to the text's `magnitude` (so longer text blocks may have greater magnitudes).
- `language` contains the language of the document, either passed in the initial request, or automatically detected if absent.
- `sentences` contains a list of the sentences extracted from the original document, which contains:
 - `sentiment` contains the *sentence level* sentiment values attached to each sentence, which contain `score` and `magnitude` values as described above.

The chart below shows some sample values and how to interpret them:

Sentiment	Sample Values
Clearly Positive*	"score": 0.8, "magnitude": 3.0
Clearly Negative*	"score": -0.6, "magnitude": 4.0
Neutral	"score": 0.1, "magnitude": 0.0
Mixed	"score": 0.0, "magnitude": 4.0

* "Clearly positive" and "clearly negative" sentiment varies for different use cases and customers. You might find differing results for your specific scenario. We recommend that you define a threshold that works for you, and then adjust the threshold after testing and verifying the results. For example, you may define a threshold of any score over 0.25 as clearly positive, and then modify the score threshold to 0.15 after reviewing your data and results and finding that scores from 0.15-0.25 should be considered positive as well.

1.2 INTELLIGENT AGENT USING CLOUD AI : IPA USE CASE 3

Analyse videos

- Virtual worker obtains a video of interest.
- Virtual worker calls cloud AI API to conduct video entities detection and video scene/shot change detection.

```
1 https://ca.yahoo.com
2 // automation flow files usually start with an URL to tell TagUI where to go
3 // files can also start with // for comments, or no URL if it's not web-related
4
5 // this flow searches github on Yahoo and captures screenshot of results
6 // for issues or questions, kindly feedback on GitHub or ksch@aisingapore.org
7 // see cheatsheet for steps, conditions, finding element identifiers, etc
8 // https://github.com/kelaberetiv/TagUI#cheat-sheet
9
10 echo 'NOTE - this sample may not work for EU users due to website changes for GDPR'
11 echo 'EU users may have to modify script to first click on announcement OK button'
12 echo ''
13
14 // lines which begin with // means user comments, which are ignored during execution
15
16 // use live step to try steps or code interactively for Chrome / visual automation
17 // this helps accelerate automation script development as you can test each step
18 // live
```

Source

https://github.com/telescopeteleuser/workshop_blog/blob/master/wechat_tool_py3_local/lesson_5_py3_local.ipynb

短片预览 / Video viewing

```
In [3]: # 多媒体文件的二进制base64码转换 (Define media pre-processing functions)

# Import the base64 encoding library.
import base64, io, sys, IPython.display

# Python 2
if sys.version_info[0] < 3:
    import urllib2
# Python 3
else:
    import urllib.request

# Pass the media data to an encoding function.
def encode_media(media_file):
    with io.open(media_file, "rb") as media_file:
        media_content = media_file.read()
# Python 2
    if sys.version_info[0] < 3:
        return base64.b64encode(media_content).decode('ascii')
# Python 3
    else:
        return base64.b64encode(media_content).decode('utf-8')
```

```
In [4]: video_file = 'reference/video_IPA.mp4'
# video_file = 'reference/SampleVideo_360x240_1mb.mp4'
# video_file = 'reference/SampleVideo_360x240_2mb.mp4'
```

```
In [5]: IPython.display.HTML(data=
    '''<video alt="test" controls><source src="data:video/mp4;base64,{0}" type="video/mp4" /></video>'''
    .format(encode_media(video_file)))
```

Out[5]:



```
In [12]: didi_segment_labels, didi_shot_labels, didi_frame_labels = didi_video_label_detection(video_file)
```

Processing video for label annotations:

Finished processing.
Video label description: toy
Segment 0: 0.0s to 5.5s
Confidence: 0.9247158169746399

Video label description: lego
Label category description: toy
Segment 0: 0.0s to 5.5s
Confidence: 0.9257180094718933

Video label description: robot
Label category description: technology
Label category description: machine
Segment 0: 0.0s to 5.5s
Confidence: 0.32479360699653625

Shot label description: toy
Segment 0: 0.0s to 5.5s
Confidence: 0.9247158169746399

Shot label description: lego
Label category description: toy
Segment 0: 0.0s to 5.5s
Confidence: 0.9257180094718933

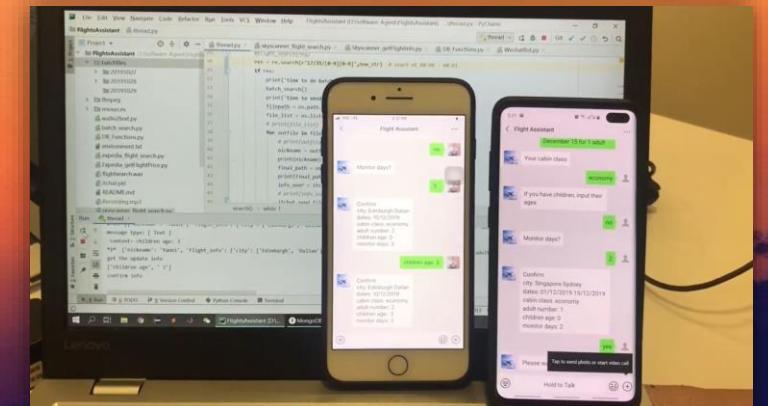
Shot label description: robot
Label category description: technology
Label category description: machine
Segment 0: 0.0s to 5.5s
Confidence: 0.32479360699653625

```
In [13]: didi_segment_labels
```

```
Out[13]: [entity {
    entity_id: "/m/0138tl"
    description: "toy"
    language_code: "en-US"
}
segments {
    segment {
        start_time_offset {
        }
        end_time_offset {
            seconds: 5
            nanos: 500000000
        }
    }
    confidence: 0.9247158169746399
},
entity {
    entity_id: "/m/04ndr"
```

1.3 INTELLIGENT AGENT SYSTEM SHOW CASE

Flights Booking Assistant

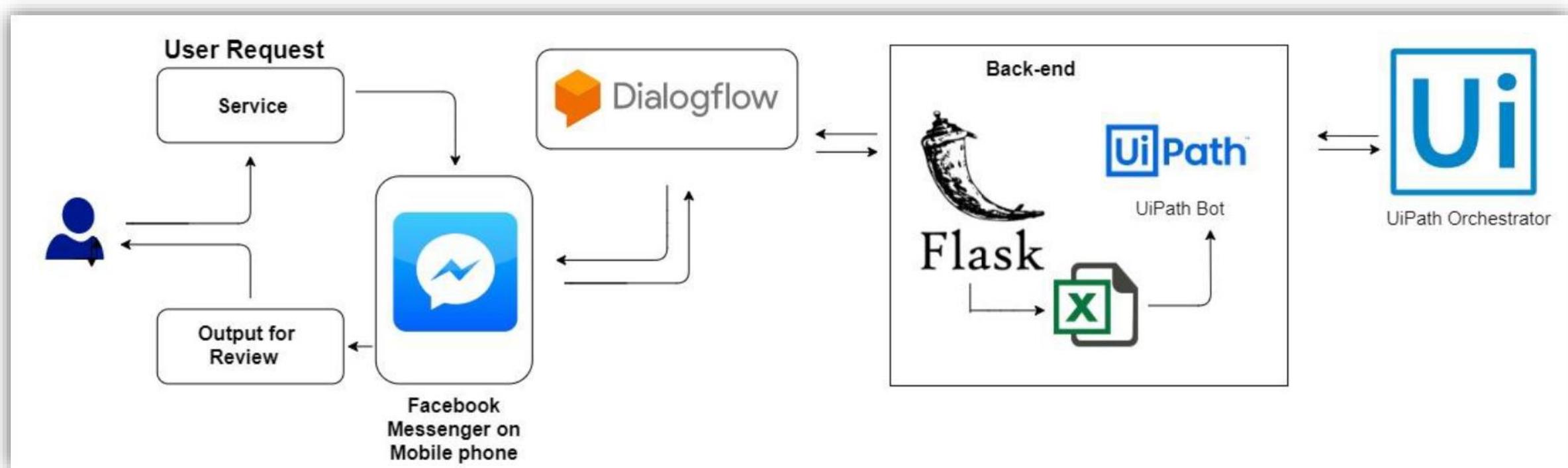


<https://youtu.be/27LIN8W5NUk>

EasyReq – Address Updates



ΕΔ\$ΥΡΕΩ



https://youtu.be/taU9_02-69o

Shoe Me The Price (SMTP) – Agent for Best Online Shoe Prices

Shoe Me
The Price



<https://youtu.be/I4g0yPRzXDo>

Intelligent Property Hunter



Property Hunter

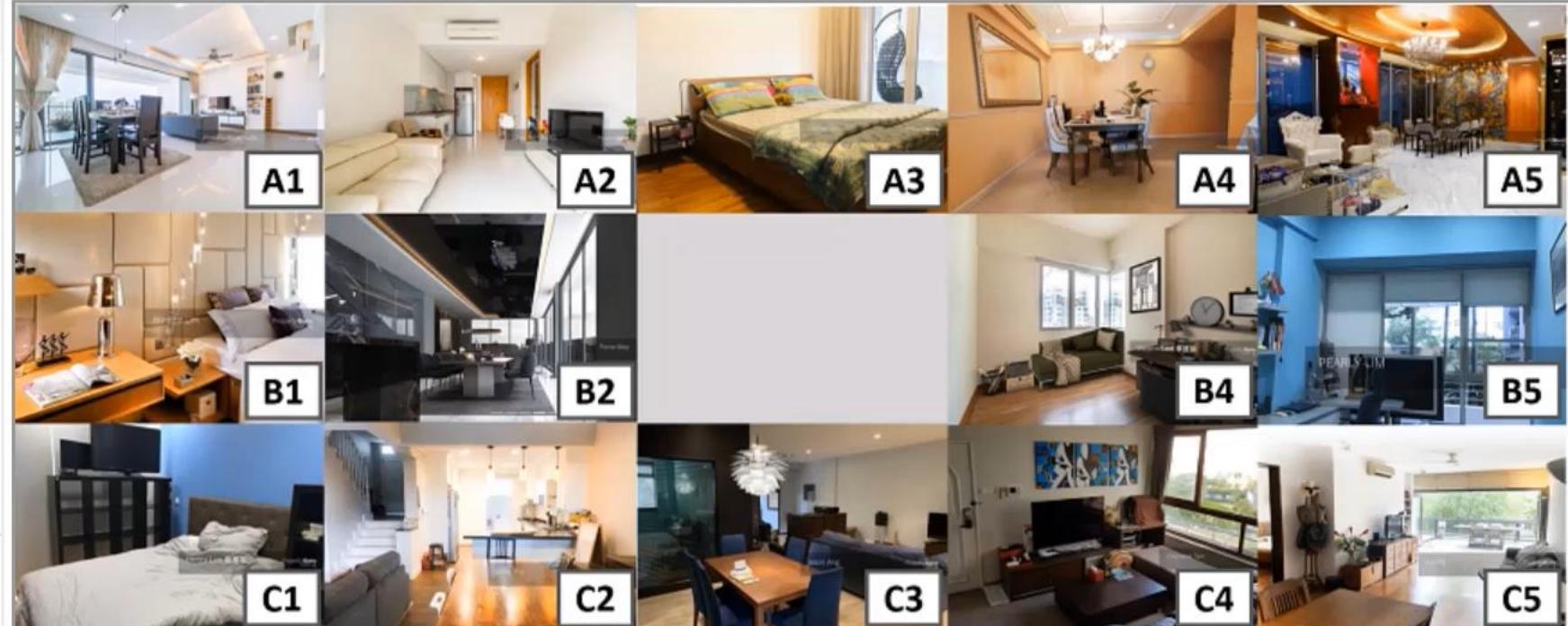
PropertyHunter

POWERED BY Dialogflow

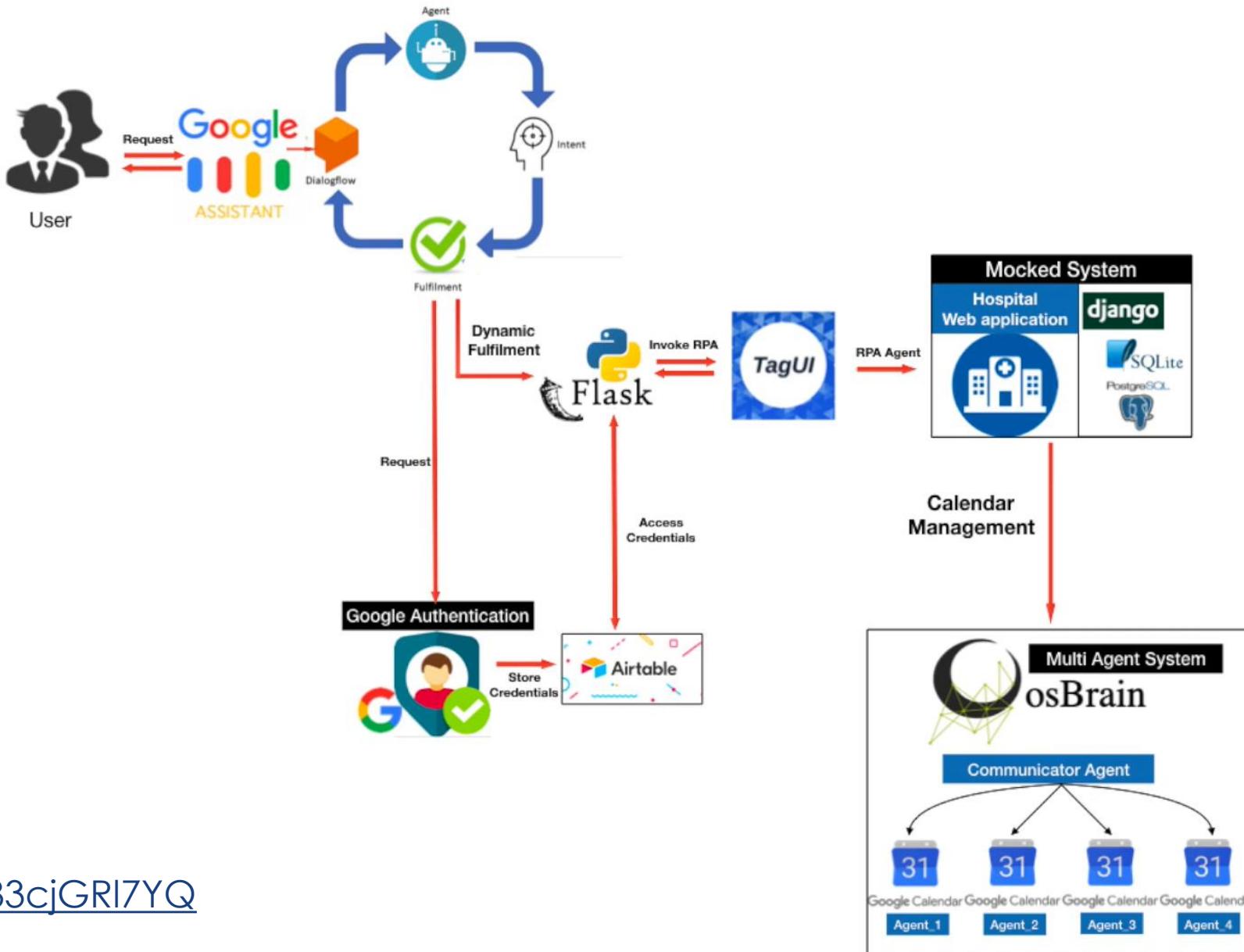
I like to view property ID12345678

Glad to hear that, just to let you know that the fair market price of this property is estimated at \$1,163,785. Shall I proceed to arrange a viewing for you?

Ask something...



CareBot – A companion you deserve!

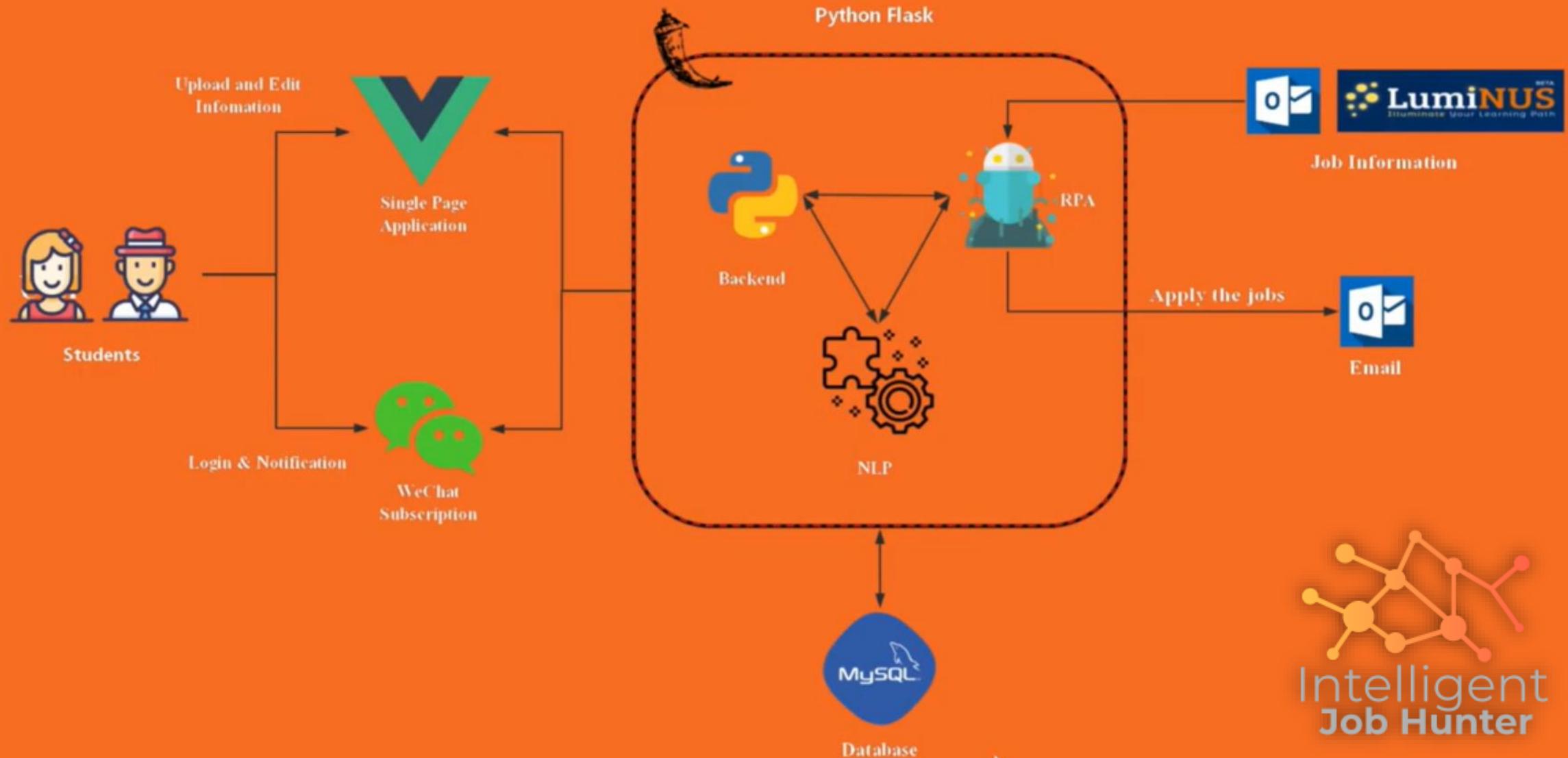


- Intelligent Systems created by past learners

<https://github.com/ISA-IPA>

1.4 INTELLIGENT AGENT WORKSHOP: VIRTUAL ASSISTANT

Intelligent Job Hunter



<https://youtu.be/PAedEATYn1E>

1.4 INTELLIGENT AGENT WORKSHOP: VIRTUAL ASSISTANT

- A job application virtual assistant, which can monitor new jobs; write tailored cover letter based on resume and job positions; and apply jobs on behalf of you, as many as you want.

1.4 INTELLIGENT AGENT WORKSHOP: VIRTUAL ASSISTANT

Individual Workshop

Job application & cover letter writing bot

- Open notebook:

https://github.com/ISA-IPA/S-IPA-Workshop/blob/master/workshop2/Virtual-Assistant-Job/IPA_VirtualAssistant4Job_v003.ipynb

The screenshot shows a Jupyter Notebook interface with the title bar 'IPA_VirtualAssistant4Job_v002'. The main content area has a blue header 'Intelligent Agent Workshop: Virtual Assistant: Job & Cover Letter'. Below it, a section titled 'Agenda:' describes the workshop's goal: 'In this workshop, we are to build a JobApplicationAssistant which will monitor "Job Opportunities" email from the provided email. Based on the user's "Resume/CV" and job description, the agent can generate tailored cover letters for job application.' It lists three main steps: 'Prepare Environment', 'Recap of web based image search', and 'Virtual Assistant for Job Application & Cover Letter', each with a corresponding bullet point. At the bottom, a section titled 'Prepare Environment' contains the first step of the agenda.

Intelligent Agent Workshop: Virtual Assistant: Job & Cover Letter

Agenda:

In this workshop, we are to build a JobApplicationAssistant which will monitor "Job Opportunities" email from the provided email. Based on the user's "Resume/CV" and job description, the agent can generate tailored cover letters for job application.

- Prepare Environment
- Recap of web based image search
- Virtual Assistant for Job Application & Cover Letter
 - Exercise 1: Get all the "job opportunities" related link from the email
 - Using Local AI Components (NLP) to generate the coverletter
 - Exercise 2: using python script sendback the application emails
 - Exercise 3: Using TagUI-Python sendback the application emails

Prepare Environment

1. Prepare system level environment

1. If you are using the ISS-VM and you also open this notebook with iss-env-py3 The environment is already prepared. Please go through the notebook to learn how to use tagui and do some demo application.

End of Lecture Notes

Contact eGL

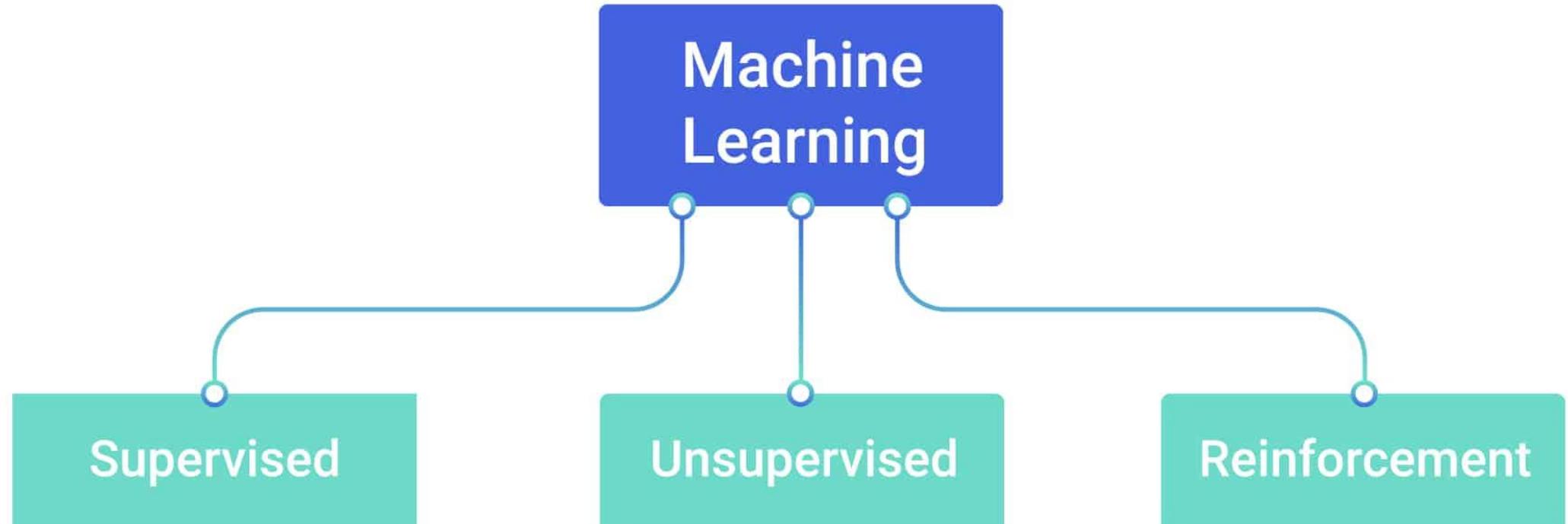
**Singapore e-Government Leadership Centre
National University of Singapore
29 Heng Mui Keng Terrace
Block D & E
Singapore 119620**

Tel : **(65) 6516 1156**
Fax : **(65) 6778 2571**
URL : **www.egl.sg**
Email : **egl-enquiries@nus.edu.sg**



Appendices

INTELLIGENT AGENT SYSTEMS USING REINFORCEMENT LEARNING (RL)



Task Driven
(Predict next value)

Type 2

knowledge discovery (white-box) by supervised algorithms, e.g. decision tree

Type 3

function approximation (black-box) by supervised algorithms, e.g. neural network; deep learning

Unsupervised

Data Driven
(Identify Clusters)

Type 1

knowledge discovery (white/black-box) by unsupervised algorithms, e.g. k-means

Reinforcement

Learn from
Mistakes/Rewards

Type 4

simulation & search based algorithms, e.g. Q-Learning, Monte Carlo tree search

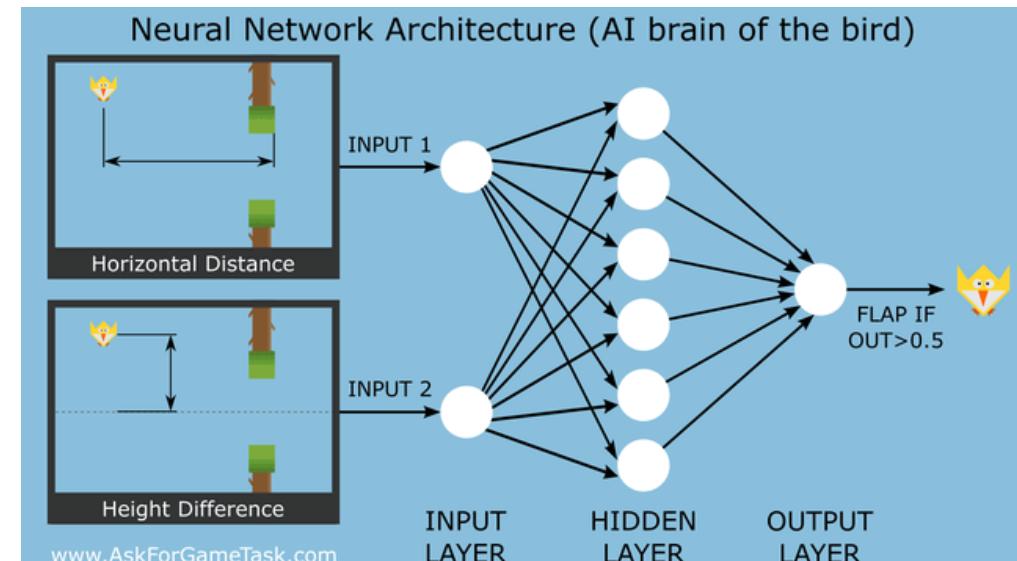
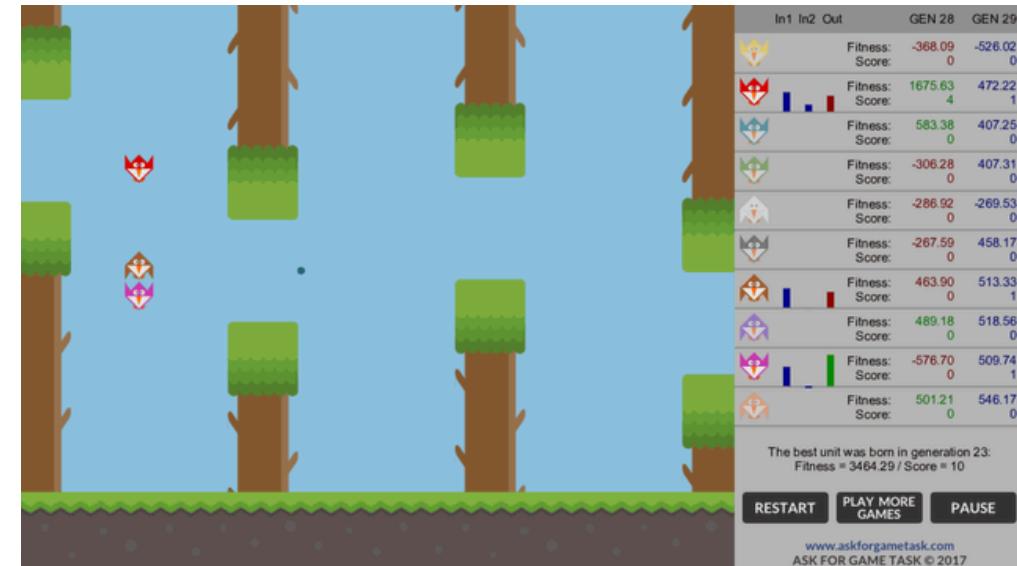
Source <https://perfectial.com/blog/reinforcement-learning-applications/>

INTELLIGENT AGENT SYSTEMS USING RL

- **Flappy Bird Agent** : The "**learning element**" is responsible for making improvements, and the "**performance element(Actor/Policy)**" is responsible for selecting external actions. The learning element uses feedback from the "**critic**", e.g. Survival time, Fitness value, Expected reward, Q function etc., on how the agent is doing and determines how the performance element should be modified to do better in the future. The performance element takes in percepts and decides on actions.

Genetic Algorithms
& MLP Neural Network

Source <https://youtu.be/aeWmdojEJf0>



- The key enabler of reinforcement learning agent is a **simulator** (a digitized virtual fast-track business/industry environment), e.g. digital twin of machinery; financial trading simulator;

INTELLIGENT AGENT WORKSHOP:

GAME AGENT

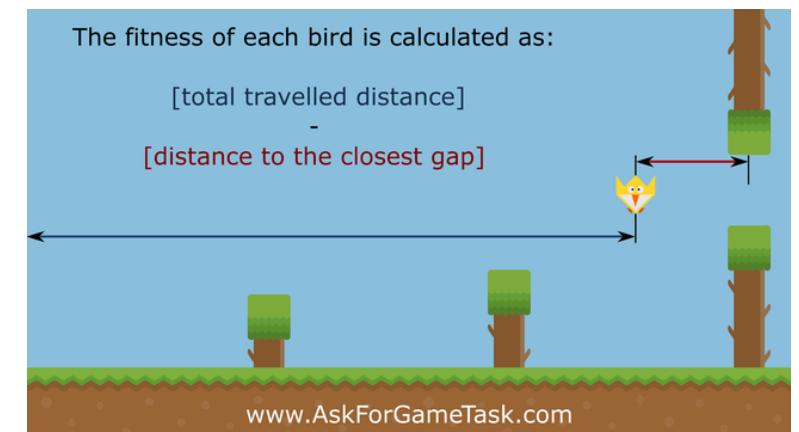
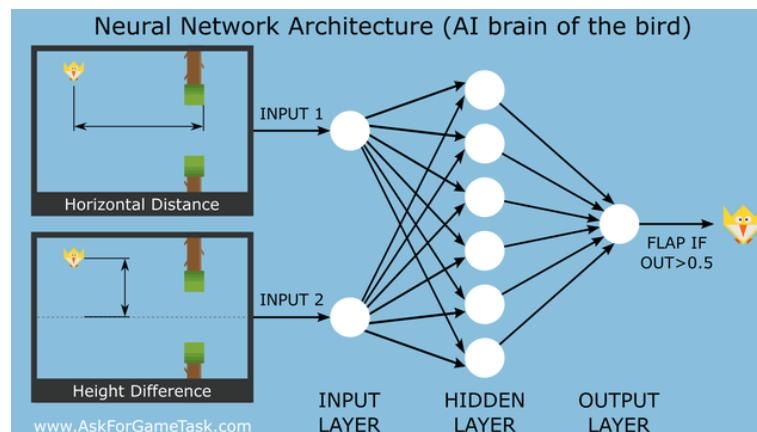
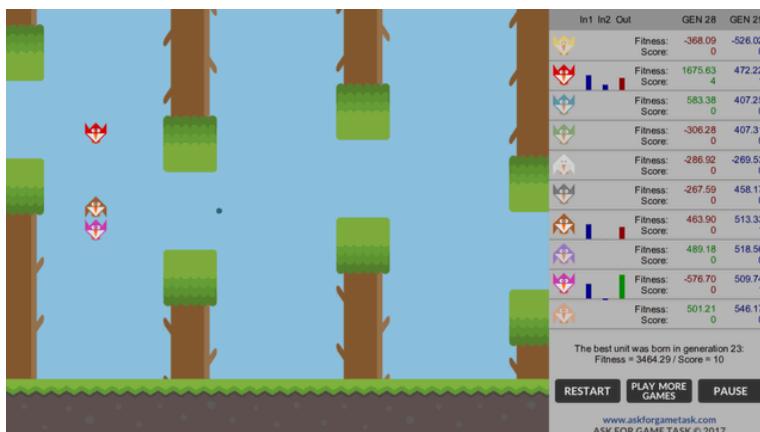
Individual Workshop

Deep dive in the game play agent for flappy bird:

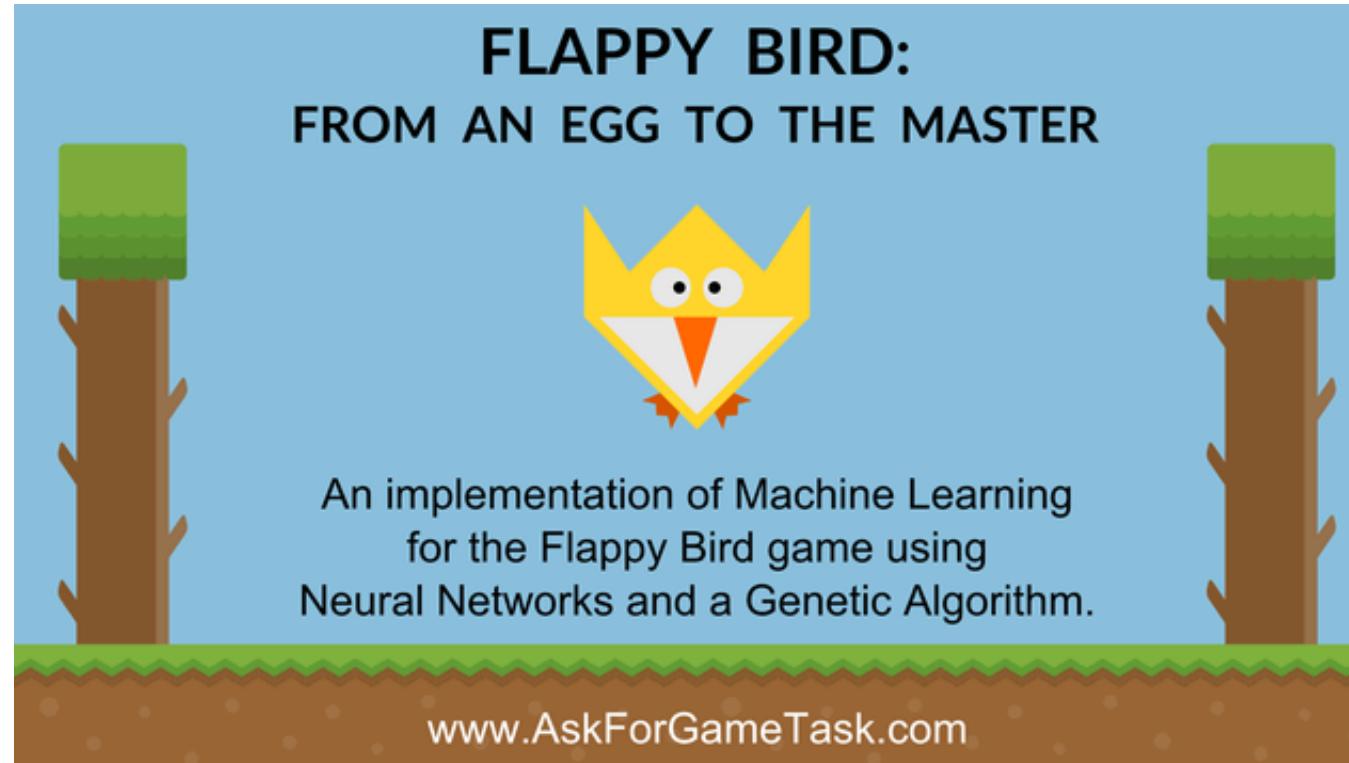
<https://github.com/telescopeuser/Machine-Learning-Flappy-Bird>

<https://www.askforgametask.com/html5/tutorials/flappy/>

<https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>



INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



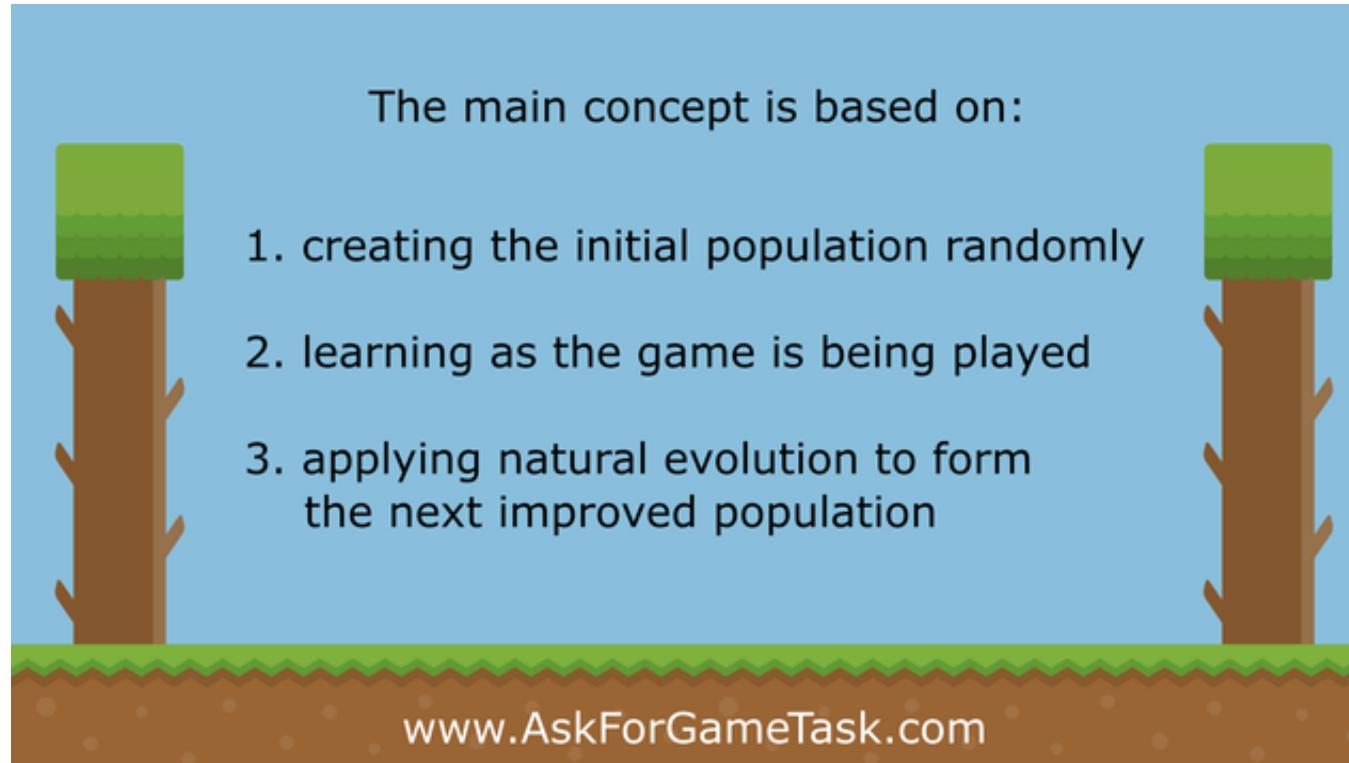
Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



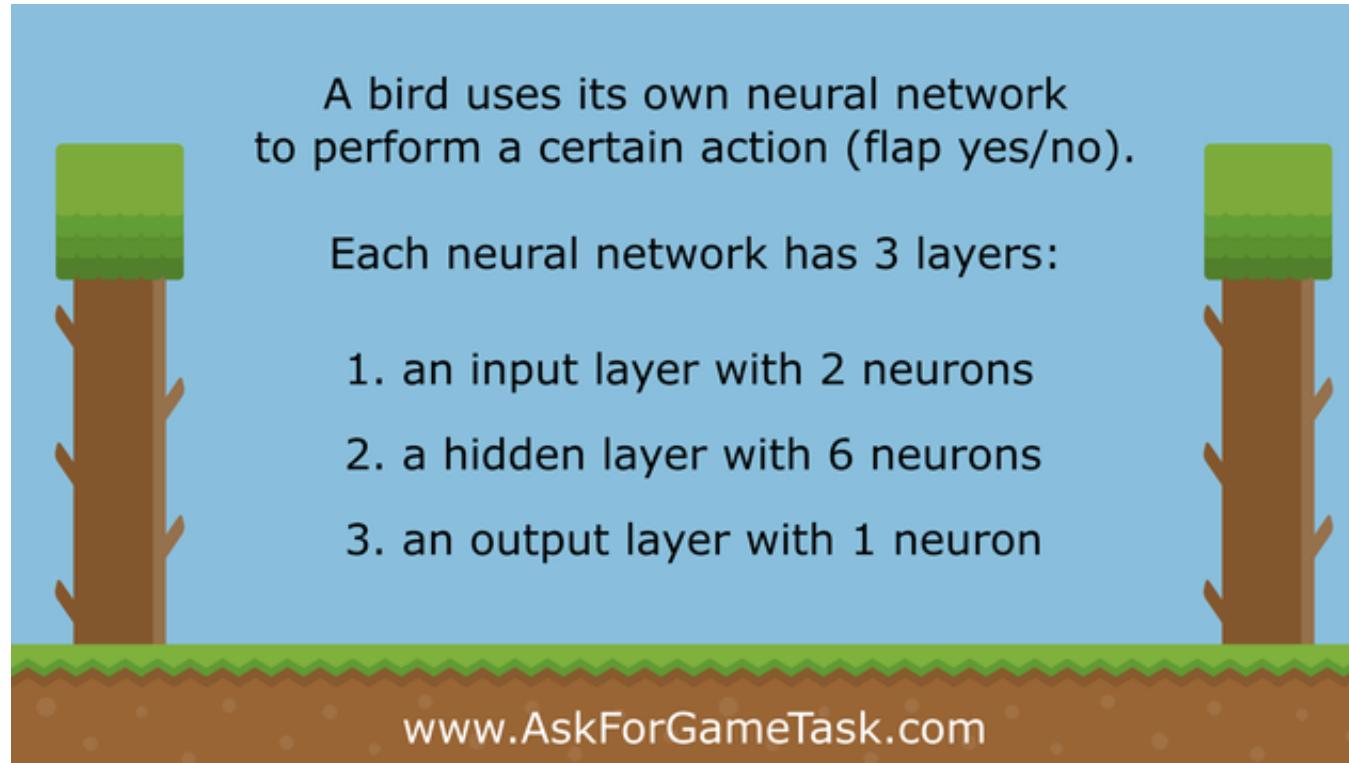
Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



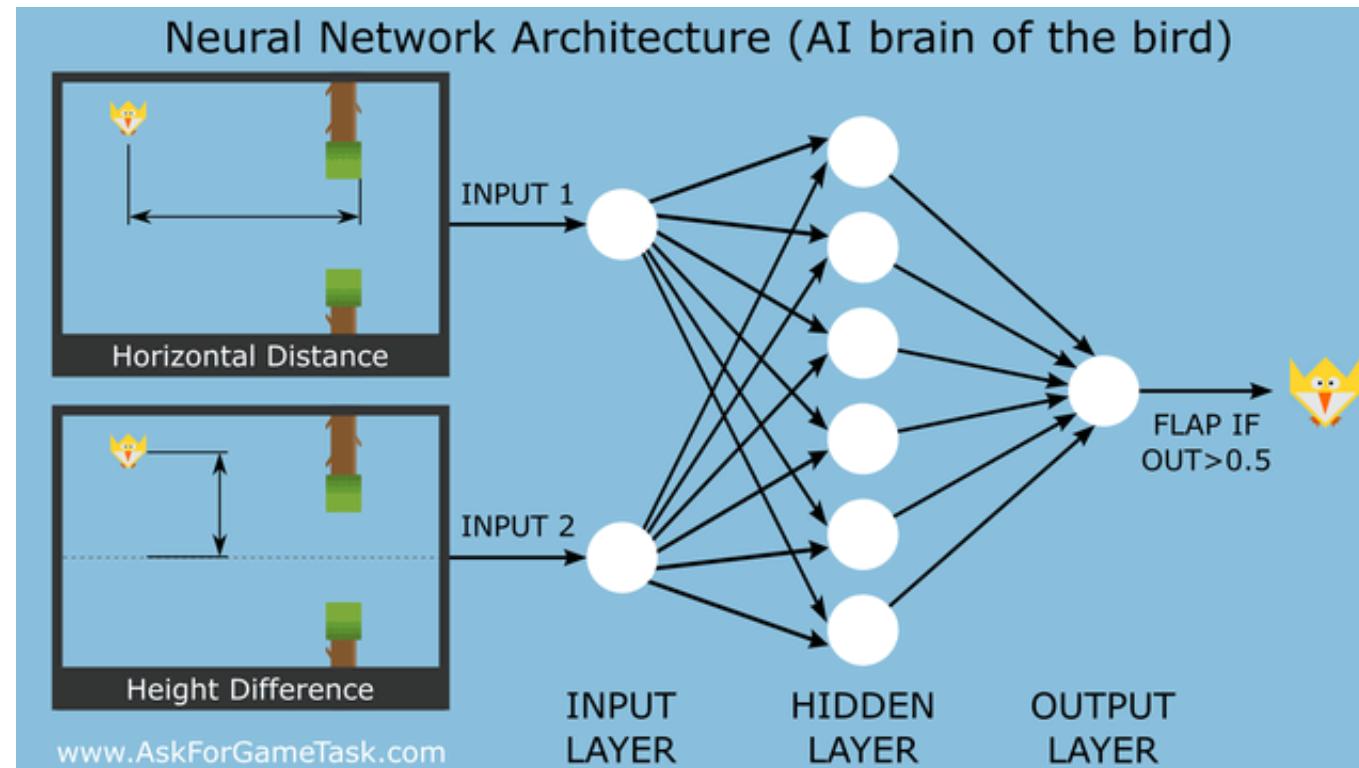
Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



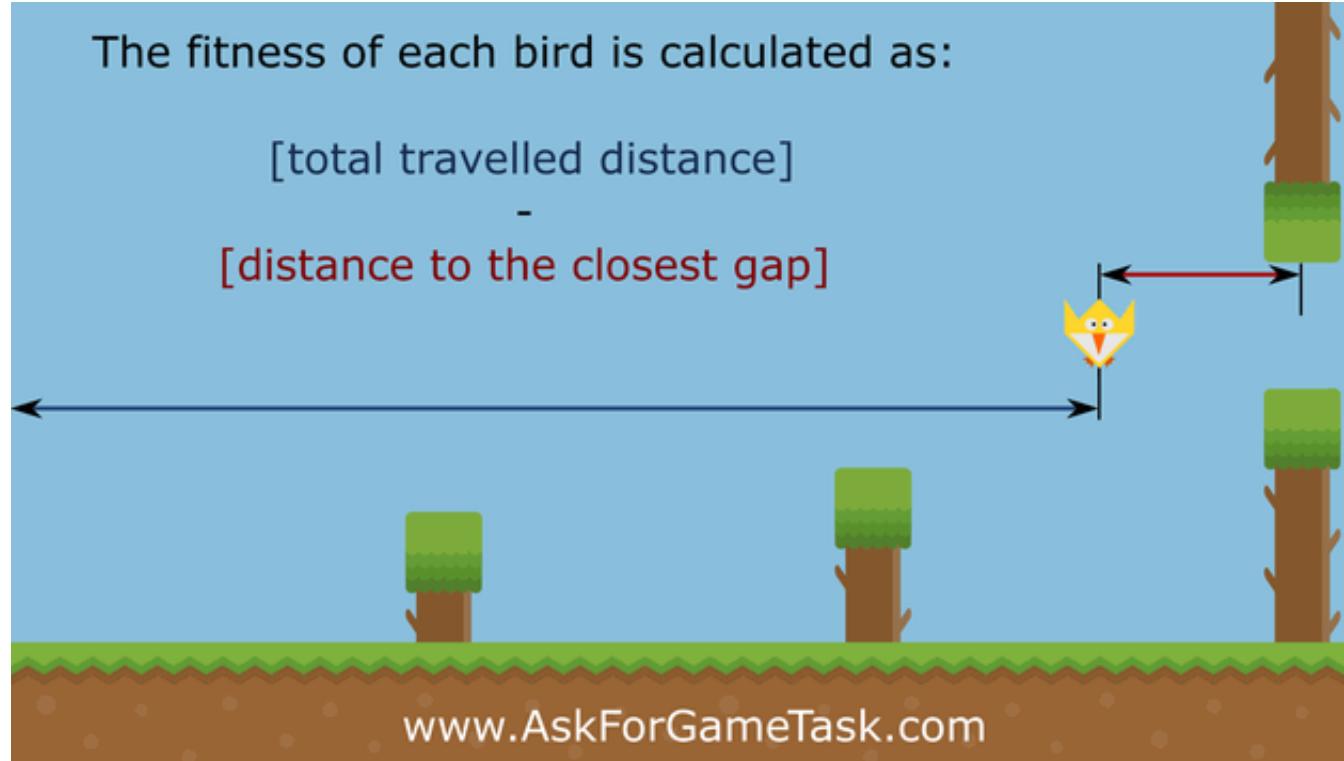
Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD

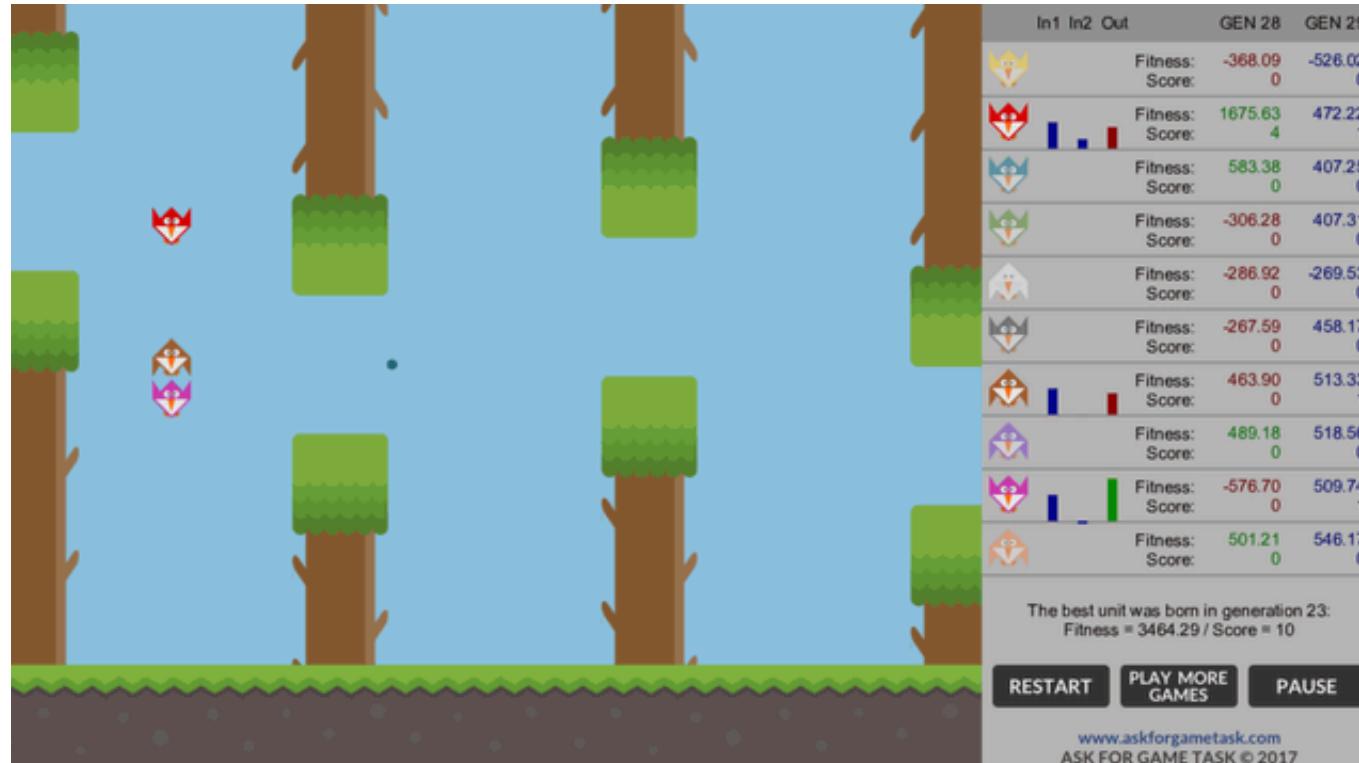
When the entire population is dead,
the fittest four birds are selected to breed a new population
by using genetic algorithm operators:

1. selection
2. crossover
3. mutation



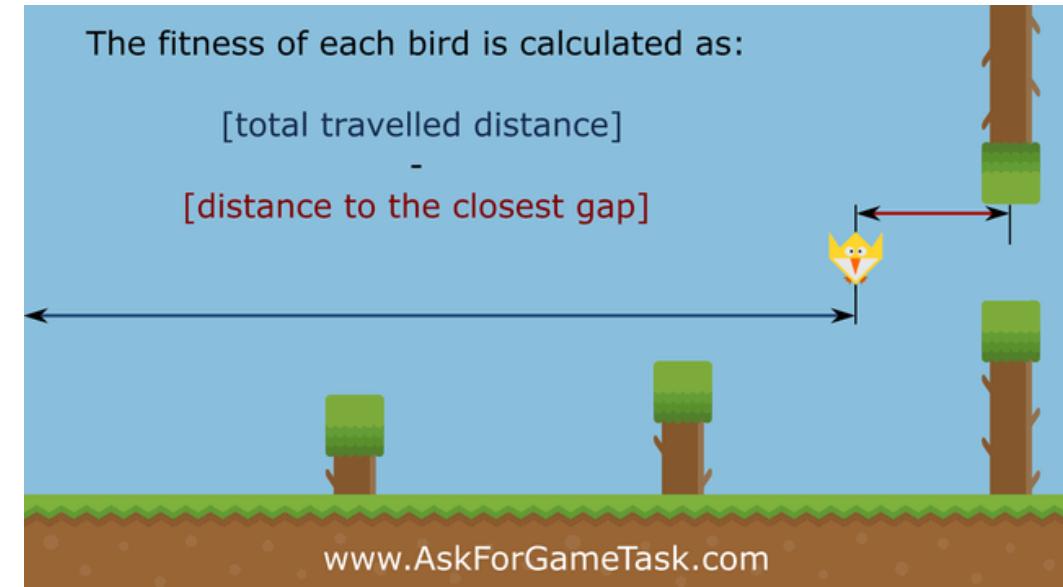
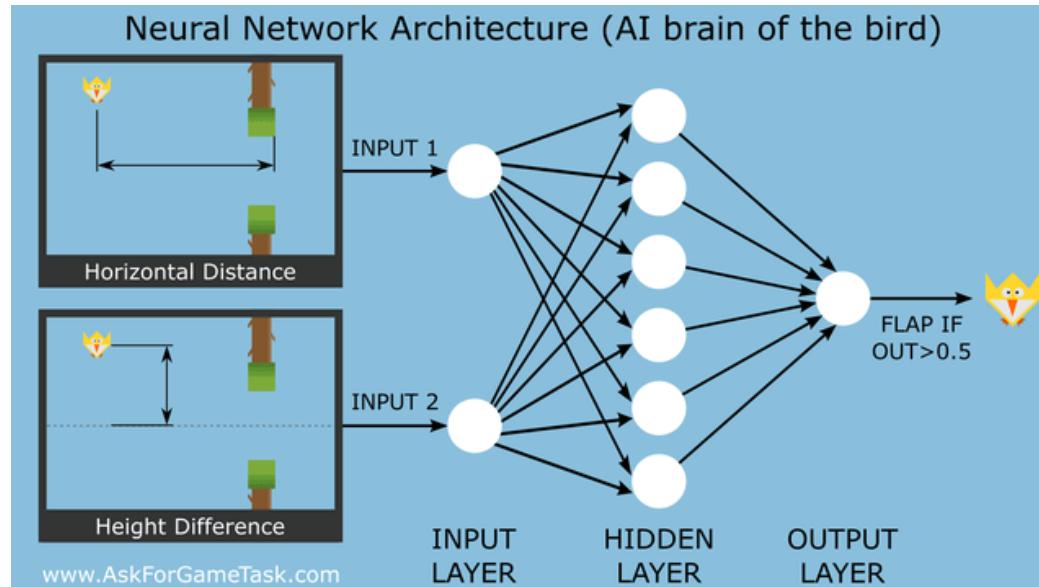
Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



Source <https://www.askforgametask.com/tutorial/machine-learning-algorithm-flappy-bird/>

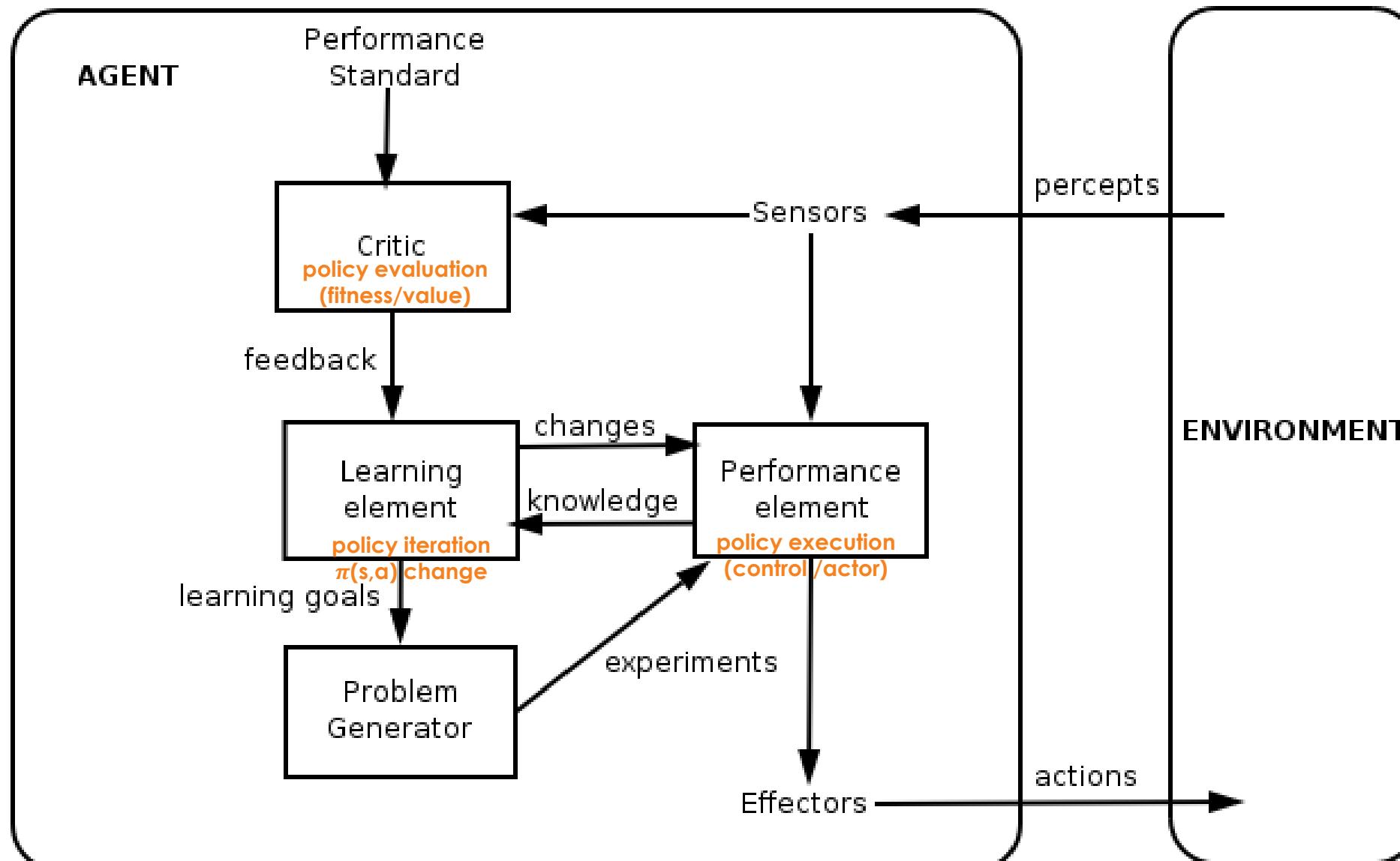
INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



A Markov decision process is a 4-tuple (S, A, P_a, R_a) , where

- S is a finite set of states,
- A is a finite set of actions (alternatively, A_s is the finite set of actions available from state s),
- $P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the probability that action a in state s at time t will lead to state s' at time $t + 1$,
- $R_a(s, s')$ is the immediate reward (or expected immediate reward) received after transitioning from state s to state s' , due to action a

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



Source https://en.wikipedia.org/wiki/Intelligent_agent

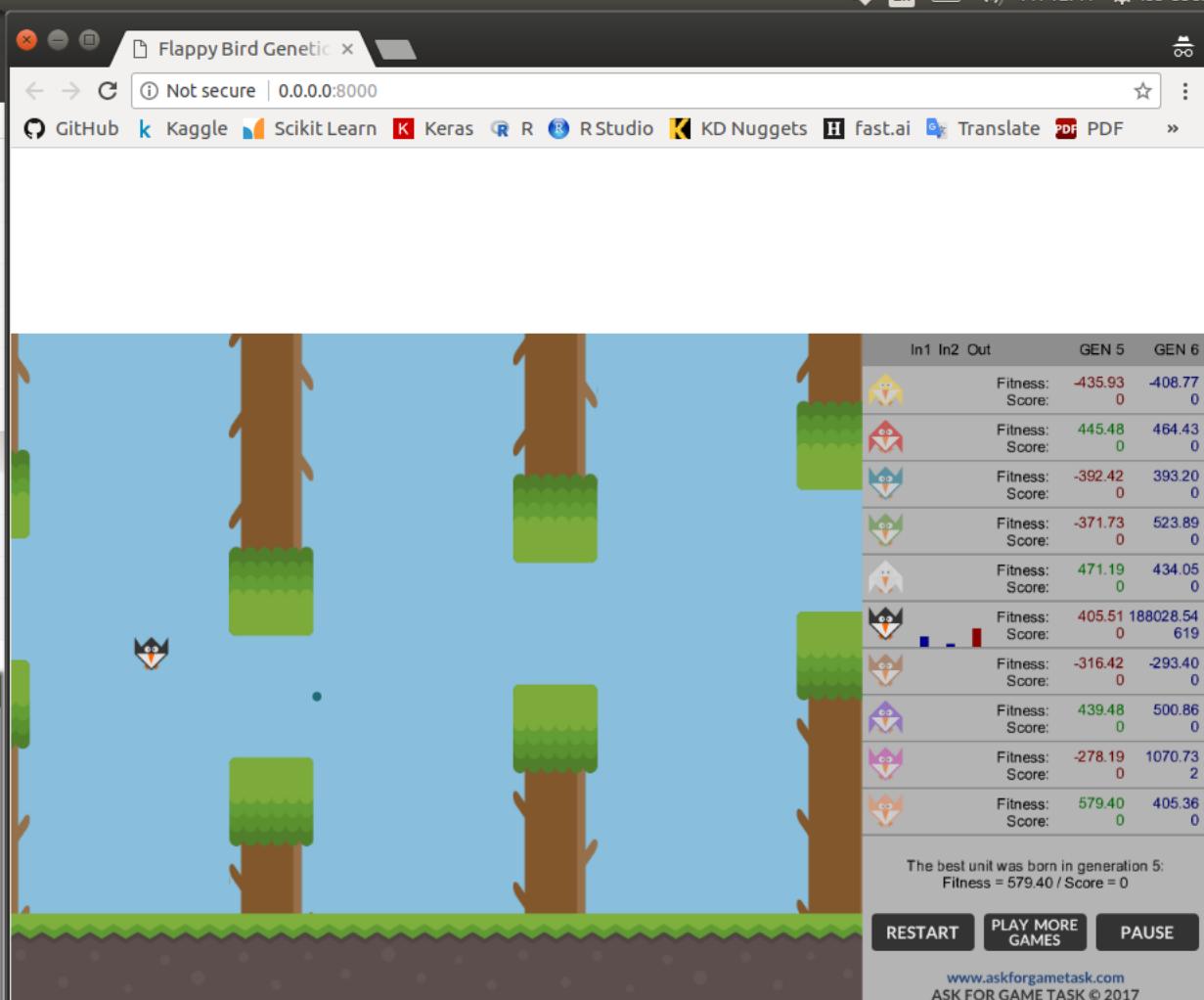
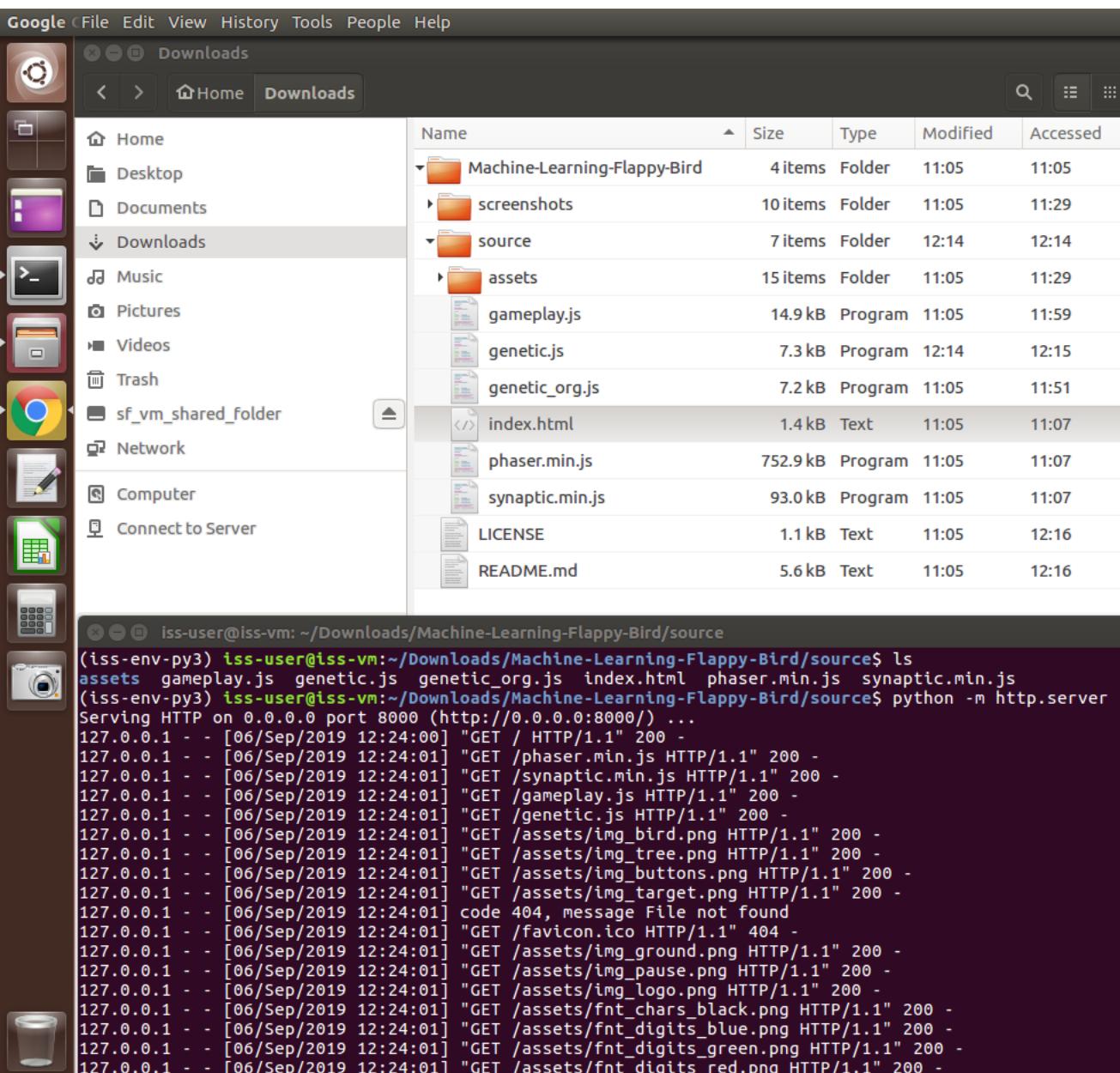
Source <https://en.wikipedia.org/wiki/File:IntelligentAgent-Learning.png>

Installation guide:

1. Download code from: <https://github.com/telescopeuser/Machine-Learning-Flappy-Bird>
2. From the directory containing ***index.html***
3. Open a terminal in **iss-vm**
4. Run: `source activate iss-env-py3`
5. Run: `python -m http.server`
6. Open browser: <http://localhost:8000>

Reference: <https://github.com/ssusnic/Machine-Learning-Flappy-Bird/issues/1>

INTELLIGENT AGENT WORKSHOP: FLAPPY BIRD



Workshop Enhancements:

- Neural Net architecture, e.g. hidden nodes, and hidden layers;
- Genetic Algorithm, e.g. population, selection, cross over, and mutation;

Observe new learning performances. Can you make the bird learn faster?

End of Appendices

Contact eGL

**Singapore e-Government Leadership Centre
National University of Singapore
29 Heng Mui Keng Terrace
Block D & E
Singapore 119620**

Tel : **(65) 6516 1156**
Fax : **(65) 6778 2571**
URL : **www.egl.sg**
Email : **egl-enquiries@nus.edu.sg**

