

ようこそPyData.Okinawaへ

Meetup 21 - OpenAI Gymで強化学習

Twitter: @PyDataOkinawa (#PyDataOkinawa)

Connpass: <http://pydataokinawa.connpass.com/>

HP: <http://pydata.okinawa>

Slack: <https://pydataokinawa.slack.com>

時間	内容	スピーカー
----	----	-------

本日の予定

時間	内容	スピーカー
08:55	開場	
09:00-09:20	PyData.Okinawaについて	PyData.Okinawa オーガナイザー
09:20-09:30	自己紹介	全員
09:30-10:00	LTセッション	ボランティア
10:00-10:45	強化学習入門	全員
14:45:-11:00	休憩	
11:00-11:50	OpenAI Gymで強化学習	全員
11:50-12:00	次回勉強会について	全員

PyData.Okinawaとは？



Creative Commons BY 2.0:

https://www.happytellus.com/gallery.php?img_id=2668

PyData.Okinawaとは？



Creative Commons BY 2.0

<http://free-photo.net/archive/entry5175.html>

PyData.Okinawaとは？

- PyData.Okinawa は Python + Data に興味のある方が交流できる沖縄を拠点にしたコミュニティです。
- データから新しい価値を生み出したいという想いがあれば、どのようなバックグラウンドを持った方でも自由に参加できるオープンなコミュニティです。
- 沖縄で定期的に開催される勉強会ではPythonをメインな言語として使いますが、機械学習やデータアナリティクスに興味を持つ方であればどのような言語を使うプログラマの方でも参加可能です。

PyData.Okinawaとは？

- またハードルは高いかもしれません、これからPythonプログラミングを覚えたい初心者の方も大歓迎です。
- 沖縄県外のコミュニティとも積極的に交流しながら、次の世代をインスピアイアするような新しい価値を生み出していくことがPyData.Okinawaの成功の姿です。

データから始まる
新しい価値を創造するコミュニティ

PyData.Tokyoとは？

Python + Data を通じて Pythonistaと データ分析者が交流するコミュニティ

2014年09月に行われたPyConJP 2014での出逢いをきっかけに、
PyData.Tokyoは生まれました。

PyConJP 2014では、データ分析や機械学習関連の発表に非常に大きな
関心が寄せられました。一方で、データの分析・可視化をするために必
要な知識や技術について、深く議論できるコミュニティやカンファレン
スがあればいいのに、という声も耳にしました。そんな想いに応えるべ
く誕生したのが、PyData.Tokyoです。

PyData.Tokyoとは？

PyData.Tokyoは、多様な分野で活躍しているPythonistaとデータ分析者がバランスよく交流できる、Python + Dataをテーマにしたコミュニティです。

- [PyData.Tokyo Official HP](#)
- [PyData.Tokyo@connpass](#)

なぜPyData.Okinawaなのか？

- データはこれからどんどん増えていく（IoT、医療、SNS、オープンデータ等）（[link](#)）
- そのデータから価値を引き出せると面白い（次の世代をインスピアイアする新しい価値の創造）
- 人とのつながりによる新しい情報や価値観との出会い
- PyData.Okinawaは実験的な場
- 沖縄県外のコミュニティとの協力体制（PyData.Tokyoなど）
- 楽しくなくなったらやめる

PyData.Okinawa News

GTA V + OpenAI Universe (2017/01/11)

OpenAI

ABOUT BLOG REQUESTS FOR RESEARCH

GTA V + Universe

CRAIG QUITER AND JACK CLARK • JANUARY 11, 2017

<https://openai.com/blog/GTA-V-plus-Universe/>

<http://gigazine.net/news/20170112-gta-v-universe-openai/>

AlphaGo behind Go Master (2017/01/05)



Demis Hassabis

@demishassabis



フォロー中

Excited to share an update on #AlphaGo!

④ 英語から翻訳



AlphaGo

04/01/17

We've been hard at work improving AlphaGo, and over the past few days we've played some unofficial online games at fast time controls with our new prototype version, to check that it's working as well as we hoped. We thank everyone who played our accounts Magister(P) and Master(P) on the Tygem and FoxGo servers, and everyone who enjoyed watching the games too! We're excited by the results and also by what we and the Go community can learn from some of the innovative and successful moves played by the new version of AlphaGo.

Having played with AlphaGo, the great grandmaster Gu Li posted that, "Together, humans and AI will soon uncover the deeper mysteries of Go". Now that our unofficial testing is complete, we're looking forward to playing some official, full-length games later this year in collaboration with Go organisations and experts, to explore the profound mysteries of the game further in this spirit of mutual enlightenment. We hope to make further announcements soon!

- 世界トップ棋士を相手に連戦連勝した謎の棋士の正体はGoogleのAlphaGo

PyMC3 (2017/01/09)

★ 固定されたツイート



Thomas Wiecki @twiecki · 1月9日

Today, after 5 years of continuous development, we released **#PyMC3** 3.0 final! Get it via pip or conda-forge.

🌐 英語から翻訳



pymc-devs/pymc3

pymc3 - Probabilistic Programming in Python: Bayesian Modeling and Probabilistic Machine Learning with Theano

github.com

← 10

153

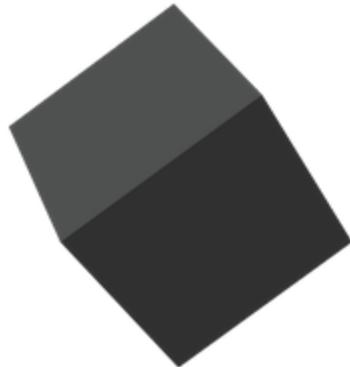
267

...

<https://github.com/pymc-devs/pymc3>

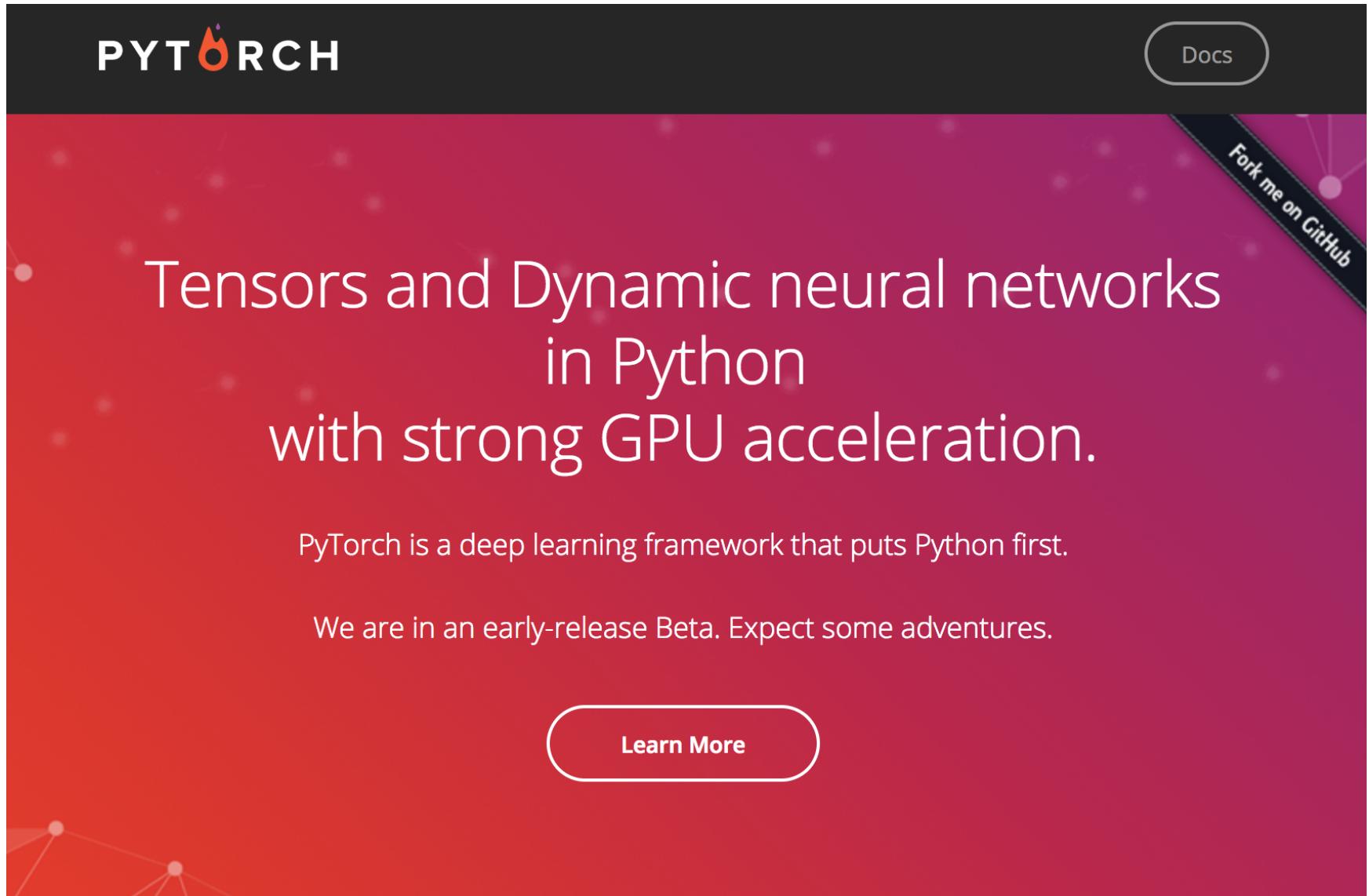
Edward

- Bayesian statistics and machine learning
- Deep learning
- Probabilistic programming



<https://github.com/blei-lab/edward>

PyTorch (2017/01/19)

A screenshot of the PyTorch landing page. The header features the PyTorch logo and a "Docs" button. The main title is "Tensors and Dynamic neural networks in Python with strong GPU acceleration." Below the title is a subtitle: "PyTorch is a deep learning framework that puts Python first." A note below that says "We are in an early-release Beta. Expect some adventures." At the bottom is a "Learn More" button.

PYTORCH

Docs

Fork me on GitHub

Tensors and Dynamic neural networks in Python with strong GPU acceleration.

PyTorch is a deep learning framework that puts Python first.

We are in an early-release Beta. Expect some adventures.

Learn More

<http://pytorch.org/>

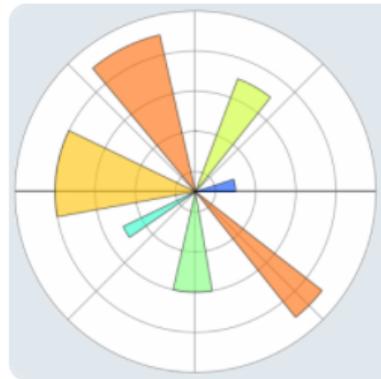
Matplotlib 2.0 (2017/01/20)



Yhat, Inc. @YhatHQ · 9時間

Matplotlib 2.0 finally released! |
[github.com/matplotlib/mat... #matplotlib](https://github.com/matplotlib/matplotlib)

🌐 英語から翻訳



matplotlib/matplotlib

matplotlib: plotting with Python. Contribute to matplotlib development by creating an account on GitHub.

github.com



35

37

<http://matplotlib.org/>

Scikit-learn Cheat Sheet (2017/01/12)

Python For Data Science Cheat Sheet

Scikit-Learn

Learn Python for data science interactively at www.DataCamp.com



Scikit-learn

Scikit-learn is an open source Python library that implements a range of machine learning, preprocessing, cross-validation and visualization algorithms using a unified interface.



A Basic Example

```
>>> from sklearn import neighbors, datasets, preprocessing
>>> from sklearn.cross_validation import train_test_split
>>> from sklearn.metrics import accuracy_score
>>> iris = datasets.load_iris()
>>> X, y = iris.data[:, :2], iris.target
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
>>> scaler = preprocessing.StandardScaler().fit(X_train)
>>> X_train = scaler.transform(X_train)
>>> X_test = scaler.transform(X_test)
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)
>>> knn.fit(X_train, y_train)
>>> y_pred = knn.predict(X_test)
>>> accuracy_score(y_test, y_pred)
```

Loading The Data

Also see NumPy & Pandas

Your data needs to be numeric and stored as NumPy arrays or SciPy sparse matrices. Other types that are convertible to numeric arrays, such as Pandas DataFrame, are also acceptable.

```
>>> import numpy as np
>>> X = np.random.random((10,5))
>>> y = np.array(['M', 'M', 'F', 'F', 'M', 'F', 'M', 'M', 'F', 'F'])
>>> X[X < 0.7] = 0
```

Training And Test Data

```
>>> from sklearn.cross_validation import train_test_split
>>> X_train, X_test, y_train, y_test = train_test_split(X,
...                                                    y,
...                                                    random_state=0)
```

Preprocessing The Data

Standardization

```
>>> from sklearn.preprocessing import StandardScaler
>>> scaler = StandardScaler().fit(X_train)
>>> standardised_X_train = scaler.transform(X_train)
>>> standardised_X_test = scaler.transform(X_test)
```

Normalization

```
>>> from sklearn.preprocessing import Normalizer
>>> normalizer = Normalizer().fit(X_train)
>>> normalized_X_train = normalizer.transform(X_train)
>>> normalized_X_test = normalizer.transform(X_test)
```

Binarization

```
>>> from sklearn.preprocessing import Binarizer
>>> binarizer = Binarizer(threshold=0.0).fit(X)
>>> binary_X = binarizer.transform(X)
```

Create Your Model

Supervised Learning Estimators

Linear Regression

```
>>> from sklearn.linear_model import LinearRegression
>>> lr = LinearRegression(normalize=True)
```

Support Vector Machines (SVM)

```
>>> from sklearn.svm import SVC
>>> svc = SVC(kernel='linear')
```

Naive Bayes

```
>>> from sklearn.naive_bayes import GaussianNB
>>> gnb = GaussianNB()
```

KNN

```
>>> from sklearn import neighbors
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)
```

Unsupervised Learning Estimators

Principal Component Analysis (PCA)

```
>>> from sklearn.decomposition import PCA
>>> pca = PCA(n_components=0.95)
```

K Means

```
>>> from sklearn.cluster import KMeans
>>> k_means = KMeans(n_clusters=3, random_state=0)
```

Model Fitting

Supervised learning

```
>>> lr.fit(X, y)
>>> knn.fit(X_train, y_train)
>>> svc.fit(X_train, y_train)
```

Fit the model to the data

Unsupervised Learning

```
>>> k_means.fit(X_train)
>>> pca_model = pca.fit_transform(X_train)
```

Fit the model to the data

Fit to data, then transform it

Prediction

Supervised Estimators

```
>>> y_pred = svc.predict(np.random.random((2,5)))
>>> y_pred = lr.predict(X_test)
>>> y_pred = knn.predict_proba(X_test)
```

Predict labels
Predict labels
Estimate probability of a label

Unsupervised Estimators

```
>>> y_pred = k_means.predict(X_test)
```

Predict labels in clustering algos

Encoding Categorical Features

```
>>> from sklearn.preprocessing import LabelEncoder
>>> enc = LabelEncoder()
>>> y = enc.fit_transform(y)
```

Imputing Missing Values

```
>>> from sklearn.preprocessing import Imputer
>>> imp = Imputer(missing_values=0, strategy='mean', axis=0)
>>> imp.fit_transform(X_train)
```

Generating Polynomial Features

```
>>> from sklearn.preprocessing import PolynomialFeatures
>>> poly = PolynomialFeatures(5)
>>> poly.fit_transform(X)
```

Evaluate Your Model's Performance

Classification Metrics

Accuracy Score

```
>>> knn.score(X_test, y_test)
>>> from sklearn.metrics import accuracy_score
>>> accuracy_score(y_test, y_pred)
```

Estimator score method
Metric scoring functions

Classification Report

```
>>> from sklearn.metrics import classification_report
>>> print(classification_report(y_test, y_pred))
```

Precision, recall, f-score
and support

Confusion Matrix

```
>>> from sklearn.metrics import confusion_matrix
>>> print(confusion_matrix(y_test, y_pred))
```

Regression Metrics

Mean Absolute Error

```
>>> y_true = [3, -0.5, 2]
>>> mean_absolute_error(y_true, y_pred)
```

Mean Squared Error

```
>>> from sklearn.metrics import mean_squared_error
>>> mean_squared_error(y_test, y_pred)
```

R² Score

```
>>> from sklearn.metrics import r2_score
```

```
>>> r2_score(y_true, y_pred)
```

R² Score

Adjusted Rand Index

```
>>> from sklearn.metrics import adjusted_rand_score
>>> adjusted_rand_score(y_true, y_pred)
```

Homogeneity

```
>>> from sklearn.metrics import homogeneity_score
>>> homogeneity_score(y_true, y_pred)
```

V-measure

```
>>> from sklearn.metrics import v_measure_score
>>> metrics.v_measure_score(y_true, y_pred)
```

Cross-Validation

```
>>> from sklearn.cross_validation import cross_val_score
>>> print(cross_val_score(knn, X_train, y_train, cv=4))
>>> print(cross_val_score(lr, X, y, cv=2))
```

Tune Your Model

Grid Search

```
>>> from sklearn.grid_search import GridSearchCV
>>> params = {"n_neighbors": np.arange(1,3),
...            "metric": ["euclidean", "cityblock"]}
>>> grid = GridSearchCV(estimator=knn,
...                      param_grid=params)
>>> grid.fit(X_train, y_train)
>>> print(grid.best_score_)
>>> print(grid.best_estimator_.n_neighbors)
```

Randomized Parameter Optimization

```
>>> from sklearn.grid_search import RandomizedSearchCV
>>> params = {"n_neighbors": range(1,5),
...            "weights": ["uniform", "distance"]}
>>> rsearch = RandomizedSearchCV(estimator=knn,
...                               param_distributions=params,
...                               cv=4,
...                               n_iter=8,
...                               random_state=5)
>>> rsearch.fit(X_train, y_train)
>>> print(rsearch.best_score_)
```



OKINAWA GPU Community vol. 1

2017/02/04 (SAT) - 2017/02/05 (SUN)



<https://docs.google.com/document/d/1ok6v3beHT49za1rm9VkaP5UDRWdiEP5jzlYbtNeUyl/edit>

PyData.Okinawaの共催

- Startup Cafe Koza
- 沖縄市

朝9時からPyData.Okinawaを開催するときには、なんと沖縄市の公共駐車場を朝8時から開けてもらえることになりました！ありがとうございます！

<https://pydataokinawa.connpass.com/event/46819/>

PyData.Okinawa基金

回	開催日	イベント名	参加人数	参加費の合計	回収済	場所	場所代	支払済	プール金	年度別累計	年度別合計
1	2015/05/30	PyData事始め	7	10500	○	Gwave Space B	3000	○	7500	7500	
2	2015/06/20	データ可視化	5	7500	○	Gwave Space B	3000	○	4500	12000	
3	2015/07/18	scikit-learn入門	9	13500	○	Gwave Space B	3000	○	10500	22500	
4	2015/08/22	モデル選択	9	13500	○	Gwave Space B	3000	○	10500	33000	
5	2015/09/19	復習ミニハッカソン	7	10500	○	Gwave Space B	3000	○	7500	40500	
6	2015/10/17	「データサイエンティスト	9	13500	○	Gwave Space B	3000	○	10500	51000	
7	2015/11/14	「データサイエンティスト	11	16500	○	Gwave Space B	3000	○	13500	64500	
8	2015/12/19	「データサイエンティスト	12	18000	○	Gwave Space A	4000	○	14000	78500	78500
9	2016/01/16	「データサイエンティスト	9	13500	○	Gwave Space 研修室	5000	○	8500	8500	
10	2016/02/20	「データサイエンティスト	8	12000	○	Gwave Space B	3000	○	9000	17500	
11	2016/03/19	「データサイエンティスト	5	7500	○	Gwave Space 研修室	5000	○	2500	20000	
12	2016/04/16	寄り合い	6	9000	○	Gwave Space B	3000	○	6000	26000	
13	2016/05/14	kaggle	9	13500	○	Gwave Space B	3000	○	10500	36500	
14	2016/06/18	kaggle	15	22500	○	Gwave Space B	3000	○	19500	56000	
15	2016/07/16	kaggle	13	19500	○	Gwave Space B	3000	○	16500	72500	
16	2016/08/20	Kerasで深層学習	18	27000	○	Gwave Space B	3000	○	24000	96500	
17	2016/09/17	Magentaで音楽生成	11	16500	○	Gwave Space 研修室	5000	○	11500	108000	
18	2016/10/15	Pandasでデータ前処理	14	21000	○	Gwave Space 研修室	5000	○	16000	124000	
19	2016/11/19	インタラクティブな可視化	15	22500	○	Startup Cafe Koza 2F	0	○	22500	146500	
20	2016/12/17	Kerasで時系列学習	19	28500	○	Startup Cafe Koza 2F	0	○	28500	175000	175000

213,500円 (2016/12/29時点)

ギークハウス沖縄プロジェクト@Campfire

あなたはパトロンになりました！

沖縄おもしろすぎるから、ギークハウス作ってみんなで集まろうよプロジェクト

👤 saboyutaka 🎙 コミュニティ



¥ 現在の支援総額
894,000円
178%
目標金額は500,000円

👤 パトロン数
96人

⌚ 募集終了まで残り
17日

<https://camp-fire.jp/projects/view/13748>



pydataokinawa

1件のパトロンになっています

現在地：沖縄県 出身地：[設定する](#) 自分のプロフィールURL：<http://camp-fire.jp/profile/pydataokinawa>
Python+Dataをテーマにした「濃い」勉強会 in Okinawa

pydataokinawa.connpass.com/
twitter.com/PyDataOkinawa

アクティビティ

お気に入り

支援したプロジェクト 1

支援中のファンクラブ 0

投稿したプロジェクト 0

メッセージ 0

設定

システム利用料について

タイトル	残り	支援日時	決済金額	選択したリターン/ お届け予定	お届け先	メッセージ	到着確認	決済状況
 沖縄おもしろすぎるから、 ギークハウス作ってみんな で集まろうよプロジェクト	33日	2016/12/29 16:49	支援金額：30,000円 システム利用料：216円	<p>□コザの町並みを見ながら屋上BBQ!(10人分) ギークハウス沖縄の屋上でBBQが出来ます。BB Qセット、食材などはこちらが用意します。 ※ギークハウス沖縄に来る必要があります。期 限1年間</p> <p>■ ステッカー ■ スマホリング ■ Tシャツ ■ スペシャルサンクスに名前を記載(サイズ大) ■ お届け予定：2017年03月</p>	確認する	連絡する		承認中

このリターンを選択中

● 30,000円以上

□コザの町並みを見ながら屋上BBQ!(10人分)

ギークハウス沖縄の屋上でBBQが出来ます。BBQセット、食材などはこちらが用意します。

※ギークハウス沖縄に来る必要があります。
期限1年間

- ステッカー
- スマホリング
- Tシャツ
- スペシャルサンクスに名前を記載(サイズ大)

 パトロ
ン: 0人

 お届け予定:
2017年03月 

PyData.Tokyo Meetup #12 - 強化学習

1月
27 PyData.Tokyo Meetup #12 -強化学習 ★

「強化学習」をテーマにした「濃い」勉強会

主催 : PyData.Tokyo



PyData.Tokyo

ハッシュタグ : #PyDataTokyo

募集内容	抽選枠	先着順
募集内容	無料	125 / 110人
運営枠	無料	4 / 6人

<https://pydatatokyo.connpass.com/event/48563/>

PythonBeginners沖縄（仮）

- Python初心者向けのワイワイもくもく会
- 7/2（土） 13:00-17:00 （第1回）
- 8/13（土） 13:00-17:00 （第2回）
- 9/10（土） 13:00-17:00 （第3回）
- 10/08（土） 13:00-17:00 （第4回）
- <http://python-beginners-okinawa.connpass.com/>
- 詳細は鈴木さんに v(^^)

PythonBeginners沖縄・助け合い所

- [Facebook ページ](#)

PyData.Okinawaの決まりごと

日時

- 定期開催（毎月第3土曜日 9:00-12:00）

参加費用

- 1500円
- 会場費 & 県外のコミュニティから人を呼んだりするため
- 飲み代ではない
- オーガナイザーも出す

その他

- 楽しく、クリエイティブに

自己紹介（ひとり1分ぐらい）

- 名前
- プロフィール（仕事・肩書きなど）

PyData.Okinawaのこれまで (1/4)

- PyData.Okinawa Meetup #1 - PyData事始め
- PyData.Okinawa Meetup #2 - データ可視化
- PyData.Okinawa Meetup #3 - scikit-learn入門
- PyData.Okinawa Meetup #4 - モデル選択
- PyData.Okinawa Meetup #5 - 復習ミニハッカソン

PyData.Okinawaのこれまで (2/4)

- PyData.Okinawa Meetup #6 - 「データサイエンティスト養成読本：機械学習入門編」読書会 1
 - 第1部 特集1 機械学習を使いたい人のための入門講座（よぎさん）
 - 第1部 特集2 機械学習の基礎知識（ひがさん）
- PyData.Okinawa Meetup #7 - 「データサイエンティスト養成読本：機械学習入門編」読書会 2
 - 第1部 特集2 機械学習の基礎知識（古川さん）
 - 第2部 特集2 Pythonによる機械学習入門（岩村さん）
- PyData.Okinawa Meetup #8 - 「データサイエンティスト養成読本：機械学習入門編」読書会 3
 - 第2部 特集2 Pythonによる機械学習入門（岩村さん）
 - 第1部 特集3 ビジネスに導入する機械学習（玉城さん）

PyData.Okinawaのこれまで (3/4)

- PyData.Okinawa Meetup #9 - 「データサイエンティスト養成読本：機械学習入門編」 読書会 4
 - 第1部 特集4 深層学習最前線 (大塚さん)
- PyData.Okinawa Meetup #10 - 「データサイエンティスト養成読本：機械学習入門編」 読書会 5
 - 第2部 特集4 Pythonで画像認識にチャレンジ (ちんさん)
- PyData.Okinawa Meetup #11 - 「データサイエンティスト養成読本：機械学習入門編」 読書会 6
 - 第2部 特集3 推薦システム入門 (玉城さん)
- PyData.Okinawa Meetup #12 - 寄り合い

PyData.Okinawaのこれまで (4/4)

- [PyData.Okinawa Meetup #13 - kaggle - Kobe Bryant Shot Selection](#)
- [PyData.Okinawa Meetup #14 - kaggle - Shelter Animal Outcomes \(Part 1\)](#)
- [PyData.Okinawa Meetup #15 - kaggle - Shelter Animal Outcomes \(Part 2\)](#)
- [PyData.Okinawa Meetup #16 - kerasで深層学習](#)
- [PyData.Okinawa Meetup #17- Magentaで音楽生成](#)
- [PyData.Okinawa Meetup #18 - Pandasでデータ前処理](#)
- [PyData.Okinawa Meetup #19 - インタラクティブな可視化](#)
- [PyData.Okinawa Meetup #20 - Kerasで時系列学習](#)
- [PyData.Okinawa 忘年会 2016](#)