

FANDOM



**Data modelling projects from an idea
to production with the help of Python**

- **Intro (me, Fandom & data)**
- **ML process with Python**
 - **Idea**
 - **Data preparation**
 - **Data modelling**
 - **Communication of the results**
 - **Implementation on production**
 - **Monitoring of model performance**
- **Questions**

Goal

- **Know which Python libraries are helpful on each step in machine learning projects**
 - Fandom set up example
- **Machine learning projects are no magic and we should not treat it like it**
 - And we can achieve good results with simple solutions

Intro

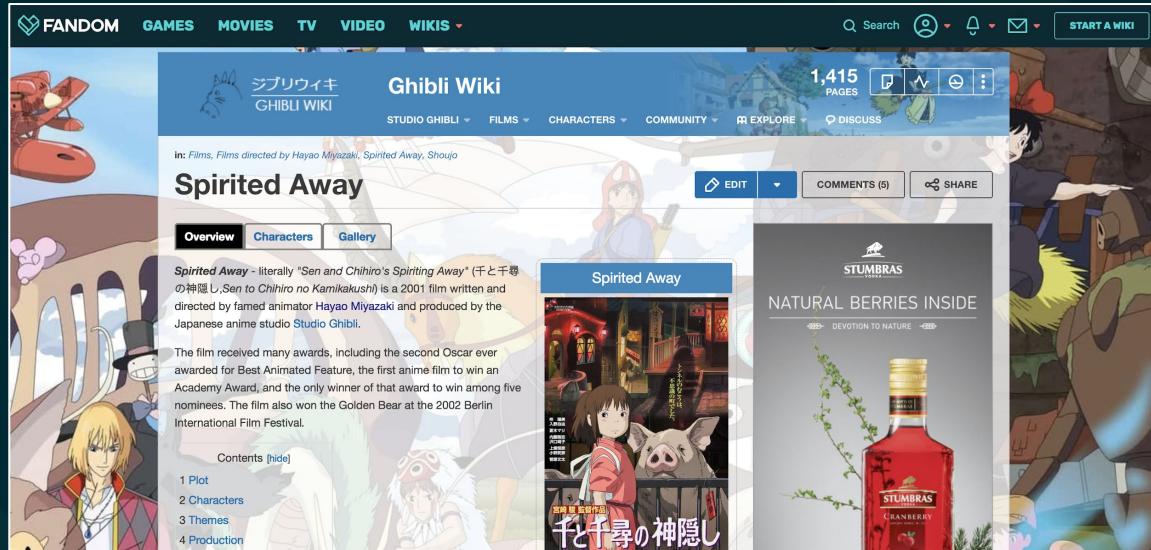
Martyna Urbanek-Trzeciak

- **MSc in Biotechnology, PhD in Biochemistry & MSc in Computer Science**
- **In FANDOM for over 2 years as data analyst (mostly with Ad Engineering team)**

Fandom

- Entertainment (mostly) wikis
- Thousands of
 - Communities
 - Pageviews
 - Users

Intro



What we like ;)

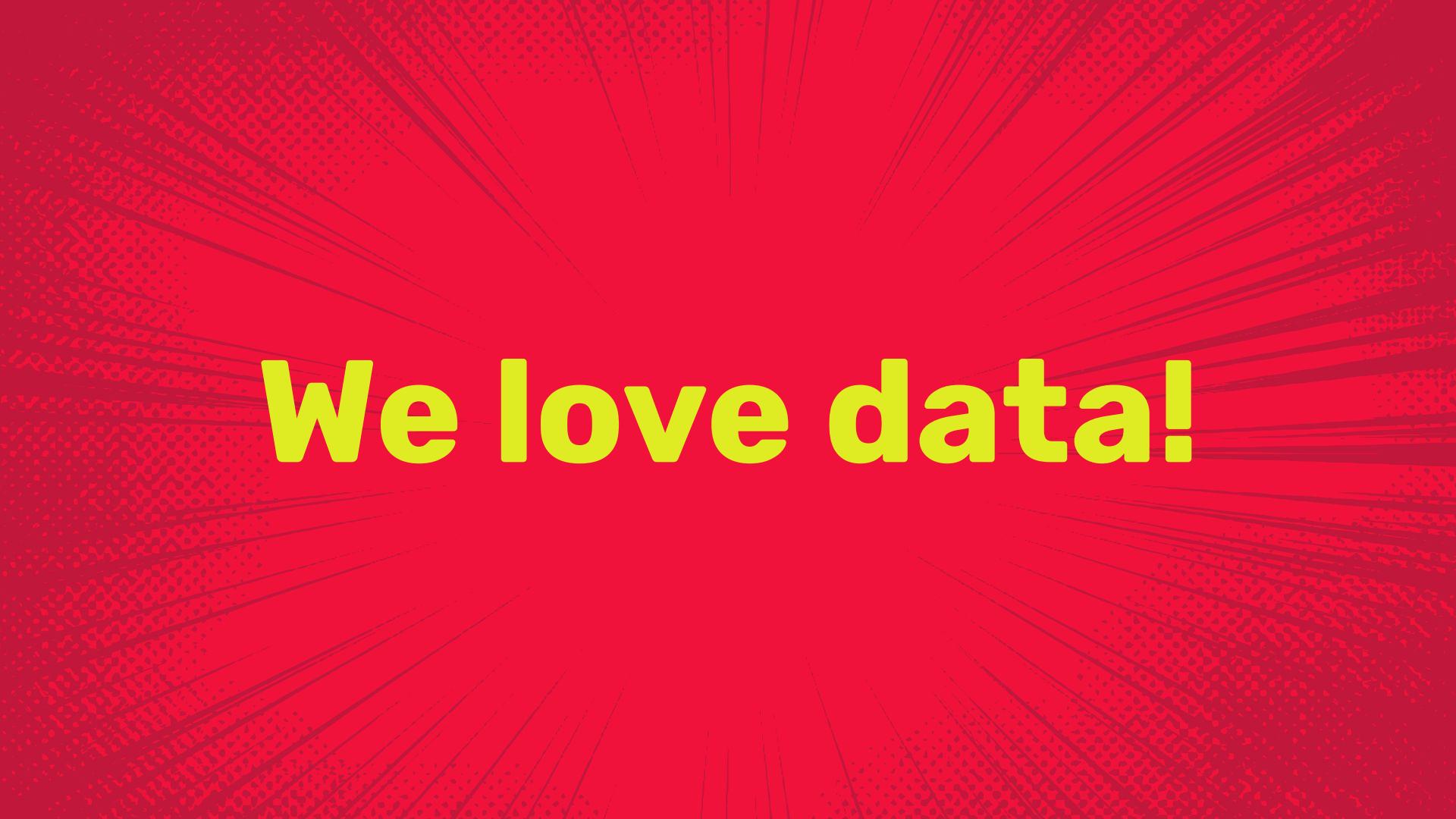
GAME OF
THRONES™

Rick and Morty

STAR
WARS

NARUTO
-ナルト-

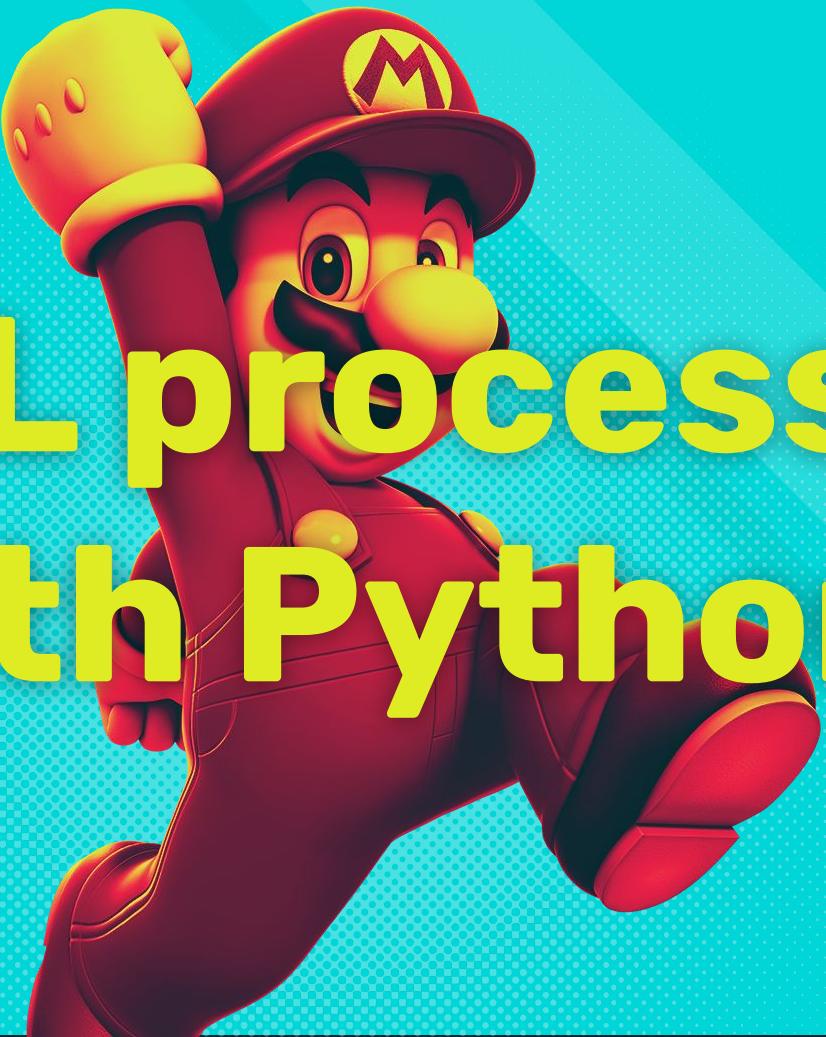
Pretty Little Liars



We love data!

Data

- **Data Warehouse**
- **Google Analytics**
- **Firebase**
- **Google Ad Manager**
- **Krux**
- **Various third party partners data**
- ...

A stylized, colorful illustration of the Mario character from the Super Mario video game series. He is depicted in his signature red color, wearing his iconic red cap with a yellow "M" emblem and a large yellow bow tie. He has a wide, smiling expression with his mouth open. The background is a vibrant blue with a subtle halftone dot pattern and several dynamic, light blue diagonal streaks that suggest motion or speed.

ML process
with Python

Idea

- During data analysis we can observe interesting trends in data and find weak points of performance that could be improved with data modelling
- Python libraries:
 - Jupyterhub
 - Pyspark3
 - Pandas, Numpy
 - Matplotlib, seaborn
 - Bokeh, plotly
 - Sroka

The screenshot shows a Jupyter Notebook interface with the following content:

```
File Edit View Insert Cell Kernel Widgets Help  
Not Trusted | PySpark3 C  
In [1]: %%configure  
{  
    "conf":  
    {  
        "spark.executor.memoryOverhead": "2G",  
        "spark.executor.memory": "5G",  
        "spark.driver.memory": "2G"  
    }  
}  
Current session config: {'conf': {'spark.executor.memoryOverhead': '2G', 'spark.executor.memory': '5G', 'spark.driver.memory': '2G'}, 'kind': 'pyspark'}  
No active sessions.  
In [2]: from pyspark.sql.functions import col, collect_list, struct, when, udf, avg, stddev  
from pyspark.sql.types import *  
Starting Spark application  
ID YARN Application ID Kind State Spark UI Driver log Current session?  
149 application_1558348165957_0544 pyspark idle Link Link ✓  
SparkSession available as 'spark'.  
In [3]: import numpy as np
```

Calculate earlier viewability in session

```
In [2]: adengevents = spark.table("statsdb.fact_adengadinfo_events").  
where((col('year')==='2019')).  
where((col('month')==='06')) & (col('day')==='01')).  
where(col('skin')==='mercury').  
where(col('bot_name')==='').  
where(col('ad_status').isin(['success'])).  
where((col('pv_unique_id') != 'undefined') & (col('pv_unique_id').isNotNull())).  
select('pv_unique_id', 'beacon', 'position', 'rv', 'wsi').  
distinct().cache()
```

```
In [3]: pageviewevents = spark.table("statsdb.fact_pageview_events").  
where((col('year')==='2019')).  
where((col('month')==='06')) & (col('day')==='01')).  
where(col('pageview_type')==='').  
where((col('pv_unique_id') != 'undefined') & (col('pv_unique_id').isNotNull())).  
select('pv_unique_id', 'beacon', 'position', 'rv', 'wsi').  
distinct().cache()
```



Sroka

Open-source Python library

Sroka (polish name of magpie)
Python open-source library
published by FANDOM enables
gathering data from different
data sources within single
Python script in the same form.

"In Italian, British and French folklore, magpies are believed to have a penchant for picking up shiny items (...). In Sweden, it is further associated with witchcraft."

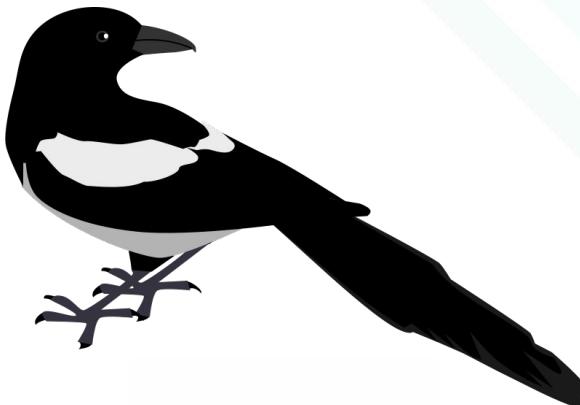
-Wikipedia

*Actually recent studies show that magpies are not attracted to shiny things

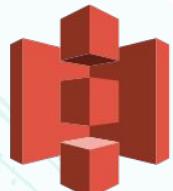


Google Ad Manager

MOAT



Google Analytics



amazon
S3



Amazon
Athena



Google
Sheets

Idea

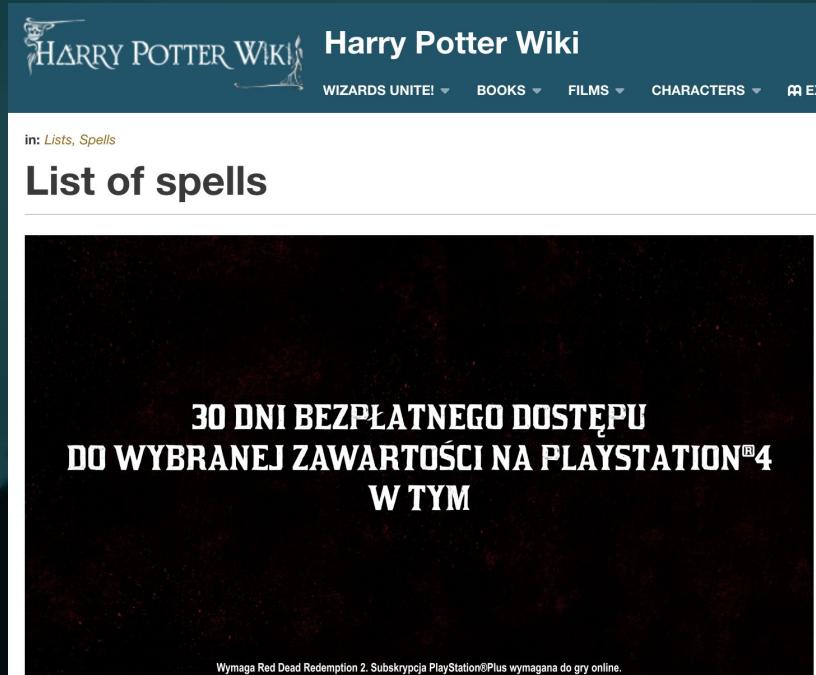
Example:

- Video ad completion rate - part of ad video plays that were played to the end.
- Trigger for analysis: Why video ad completion rate differ between ad campaigns? (besides e.g. length of ad)
- We observe that ad video completion rate differs between users, type of pages, communities, videos before which ad is played - can we capture it? Can we predict “better traffic” for video ads?

Goal

Predict “better” traffic that will have **higher video ad completion rate.**

Predict pageviews on which **user will complete video ad.**



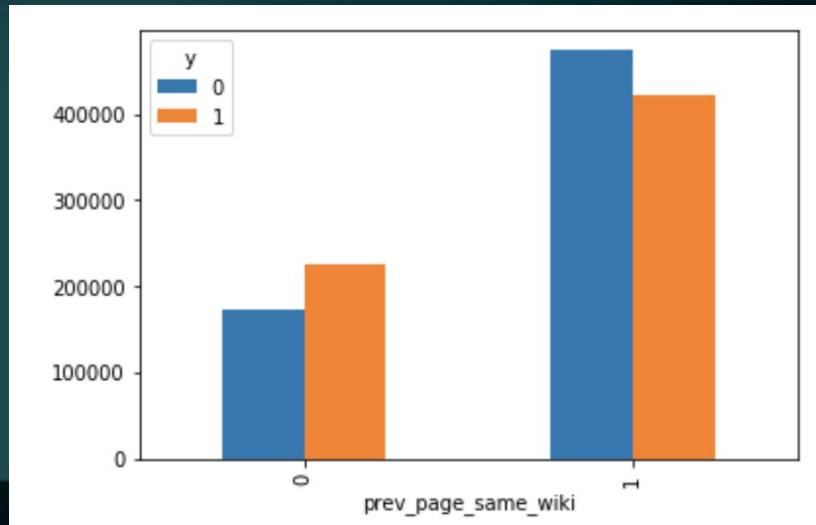
The image shows the Harry Potter Wiki homepage. At the top, there is a navigation bar with links for "WIZARDS UNITE!", "BOOKS", "FILMS", "CHARACTERS", and "EX". Below the navigation bar, there is a search bar with the placeholder "Search Harry Potter Wiki". The main content area features a large, dark rectangular overlay containing white text. The text reads: "30 DNI BEZPŁATNEGO DOSTĘPU DO WYBRANEJ ZAWARTOŚCI NA PLAYSTATION®4 W TYM". At the bottom of the page, there is a small note: "Wymaga Red Dead Redemption 2. Subskrypcja PlayStation®Plus wymagana do gry online."

Data preparation

- What features should be included?
- How features need to look like based on the algorithm we want to try out?
 - Will it be usable if we want to try different algorithms?
- Do we have all information we need
 - What simple features should we use?
 - Should we create new complex features?
 - Should we implement new events for feature creation?

Data preparation

- Python libraries:
 - Jupyterhub
 - Pyspark3
 - Pandas, Numpy
 - Matplotlib
 - Sroka
 - Sklearn
-



```
In [35]: selector = SelectKBest(chi2, k=50).fit(X_train, y_train)
X = selector.transform(X_train)
indices = selector.get_support(indices=True)
X_test_new = X_test.iloc[:, indices]
```

Data modelling

- Are we able to find a pattern?
- Can we predict outcome correctly?
- Python libraries:
 - Sklearn
 - Tensorflow
 - PyTorch
 - ...
-

```
In [31]: model = LogisticRegression(fit_intercept=True, random_state=24, class_weight='balanced', C=100,
                                  max_iter=10000)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred, digits=5))
print(confusion_matrix(y_test, y_pred))
```

Communication of the results

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	-----	-----	-----

	<i>Class 1 Predicted</i>	<i>Class 2 Predicted</i>
<i>Class 1 Actual</i>	TP	FN
<i>Class 2 Actual</i>	FP	TN

Predicted Values	Actual Values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN
	Not Pregnant	Pregnant
Positive Test Result	False Positive	True Positive
Negative Test Result	True Negative	False Negative

<https://jovianlin.io/confusion-matrix/>

<https://towardsdatascience.com/taking-the-confusion-out-of-confusion-matrices-c1ce054b3d3e>

<https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>

<https://whatis.techtarget.com/definition/confusion-matrix>

Communication of the results

		Predicted: NO
n=165		
Actual: NO		TN = 50
	Actual: YES

Class 1 Actual
Class 2 Actual



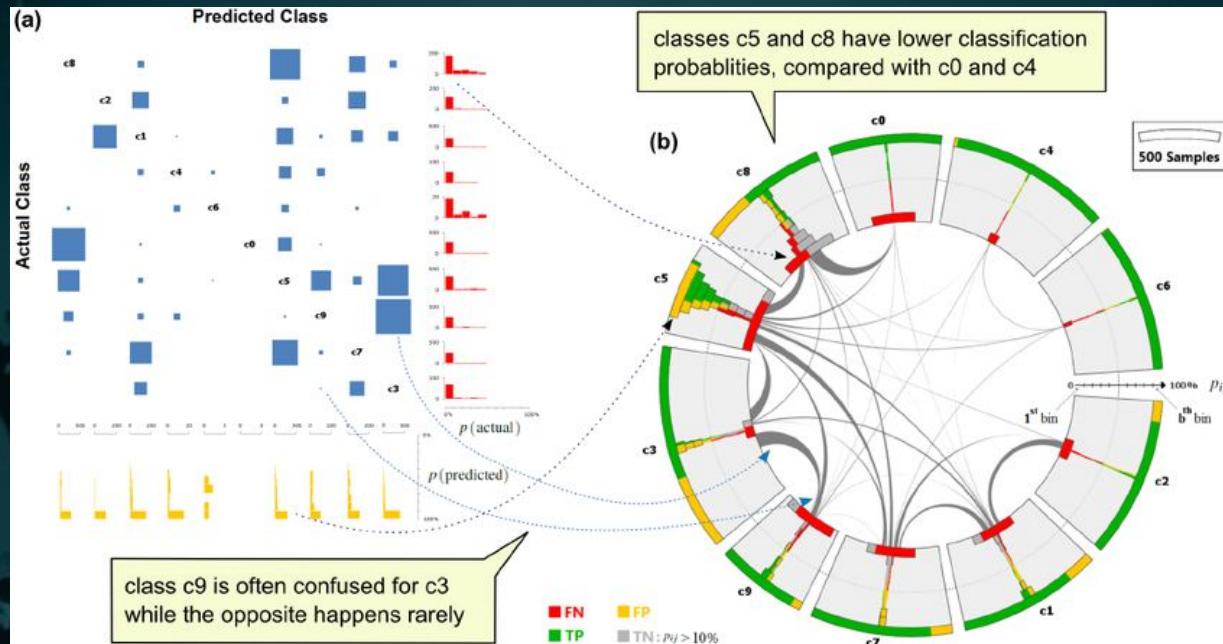
	Pregnant
	True Positive
	False Negative

<https://jovianlin.io/confusion-matrix/>

<https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>

<https://whatis.techtarget.com/definition/confusion-matrix>

Communication of the results



Communication of the results

Do we have a success? - Business interpretation

We divide our video watching users into two groups, based on our prediction:

- people are likely to watch ad (52% people, with 62% VCR)
- people probably won't watch an ad (48% people, with 52% VCR)

Initially was 57,5% VCR, so we improved it by 4,5%. VCR difference between groups is 10 pp.

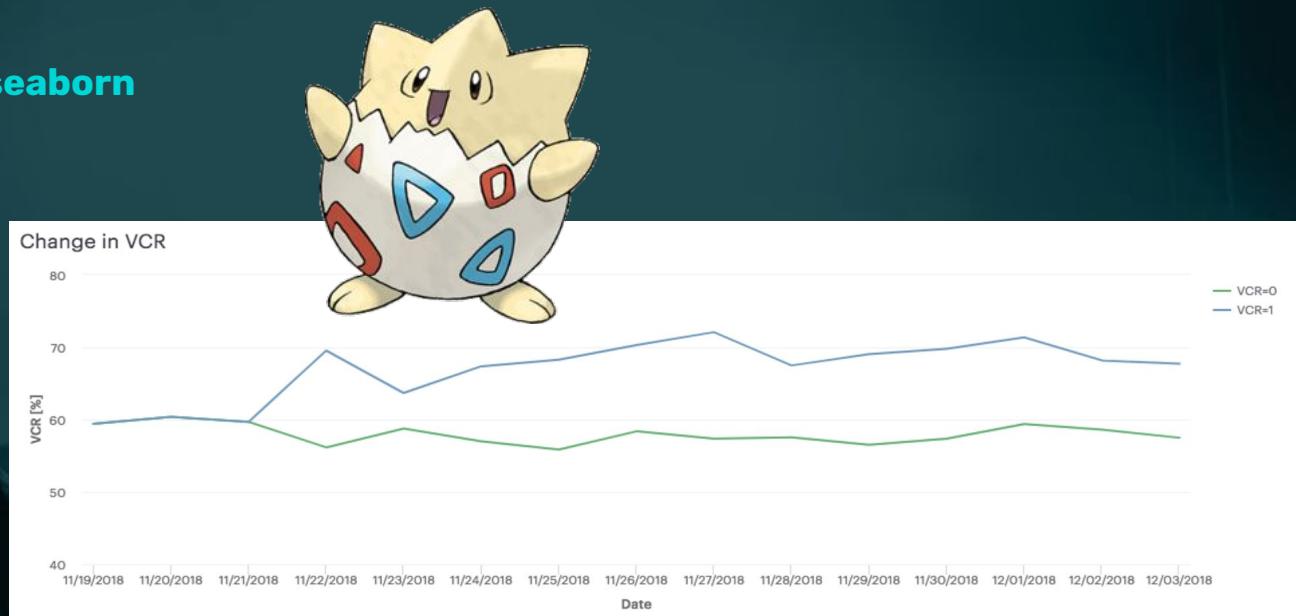
Communication of the results

No one likes a black box. We would like to know most important features and (if possible) how they could have influence our metric.

- E.g. most important feature would be if user had earlier completed ads (she is more likely to see an video ad to the end)

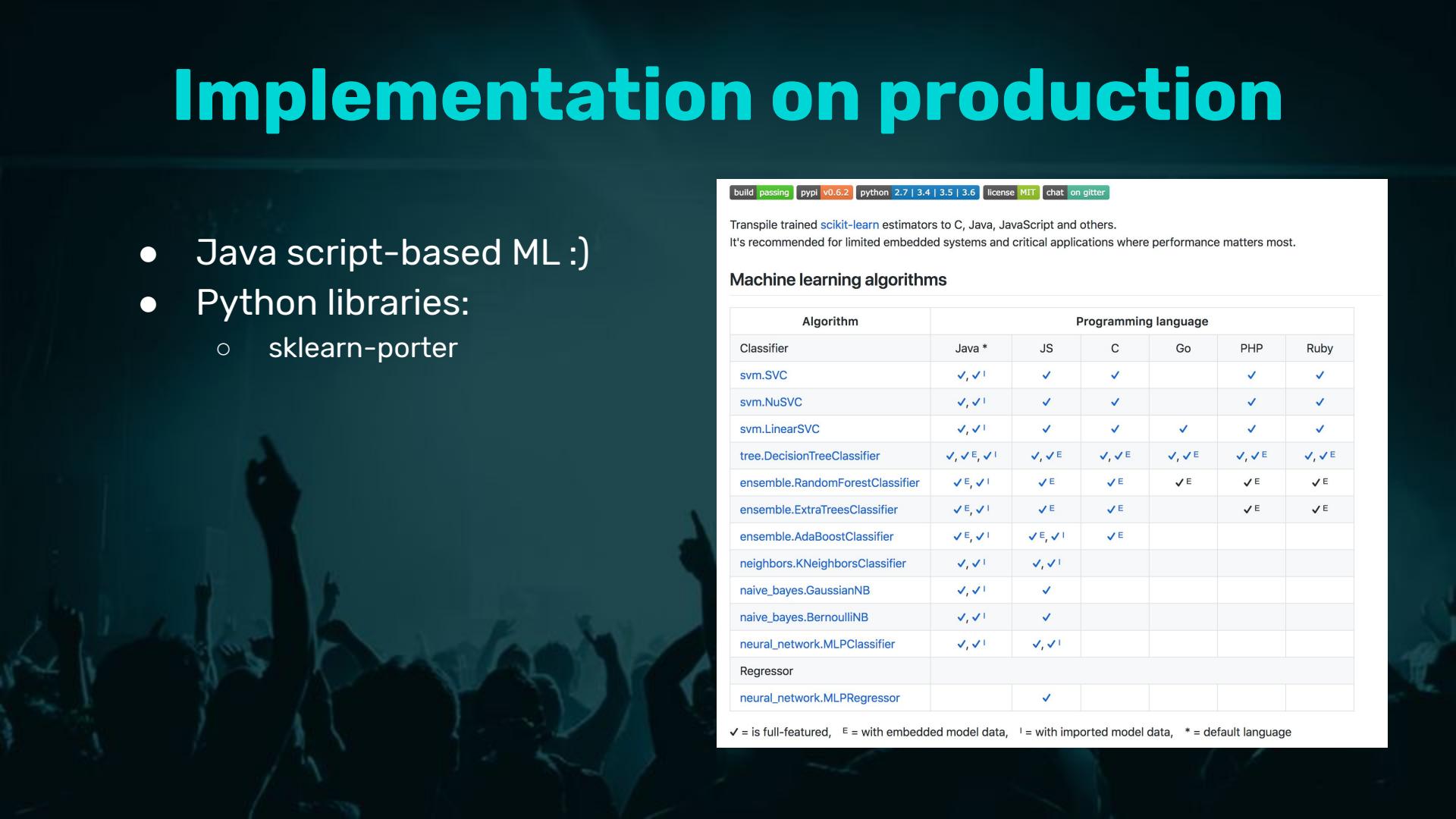
Communication of the results

- Python libraries:
 - Jupyterhub
 - Pandas
 - **Matplotlib, seaborn**
 - Sroka



Implementation on production

- Java script-based ML :)
- Python libraries:
 - sklearn-porter

A dark, semi-transparent silhouette of a crowd of people with their hands raised, visible against a lighter background.

Machine learning algorithms						
Algorithm	Programming language					
Classifier	Java *	JS	C	Go	PHP	Ruby
svm.SVC	✓, ✓ ^I	✓	✓		✓	✓
svm.NuSVC	✓, ✓ ^I	✓	✓		✓	✓
svm.LinearSVC	✓, ✓ ^I	✓	✓	✓	✓	✓
tree.DecisionTreeClassifier	✓, ✓ ^E , ✓ ^I	✓, ✓ ^E	✓, ✓ ^E	✓, ✓ ^E	✓, ✓ ^E	✓, ✓ ^E
ensemble.RandomForestClassifier	✓ ^E , ✓ ^I	✓ ^E	✓ ^E	✓ ^E	✓ ^E	✓ ^E
ensemble.ExtraTreesClassifier	✓ ^E , ✓ ^I	✓ ^E	✓ ^E		✓ ^E	✓ ^E
ensemble.AdaBoostClassifier	✓ ^E , ✓ ^I	✓ ^E , ✓ ^I	✓ ^E			
neighbors.KNeighborsClassifier	✓, ✓ ^I	✓, ✓ ^I				
naive_bayes.GaussianNB	✓, ✓ ^I	✓				
naive_bayes.BernoulliNB	✓, ✓ ^I	✓				
neural_network.MLPClassifier	✓, ✓ ^I	✓, ✓ ^I				
Regressor						
neural_network.MLPRegressor		✓				

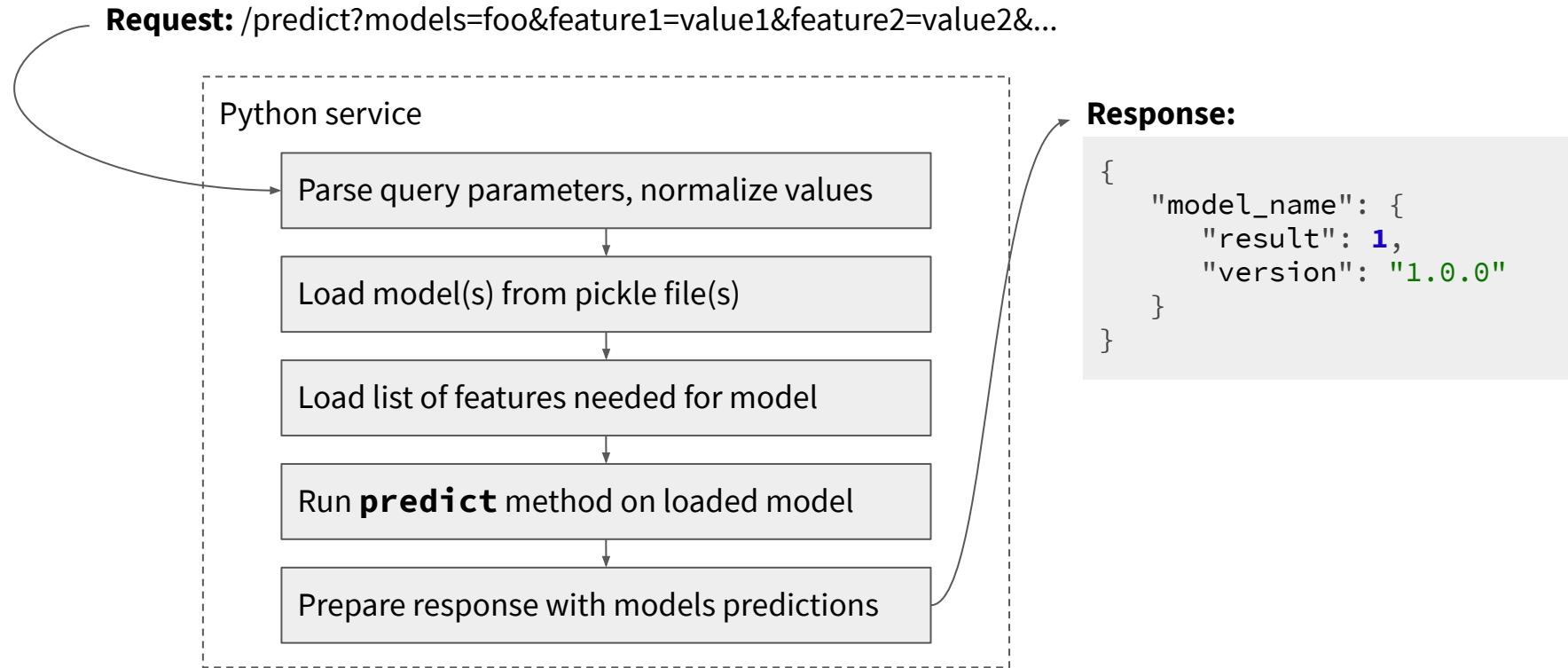
✓ = full-featured, E = with embedded model data, I = with imported model data, * = default language

Implementation on production

- Bill the lizard - runs HTTP Python server that can use trained models to predict result based on features given in query params.



Service for machine learning models



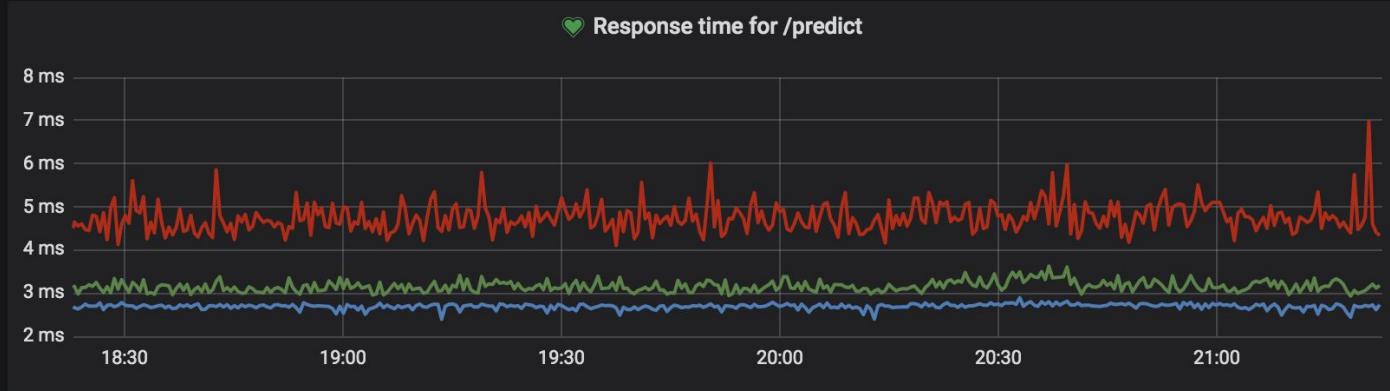
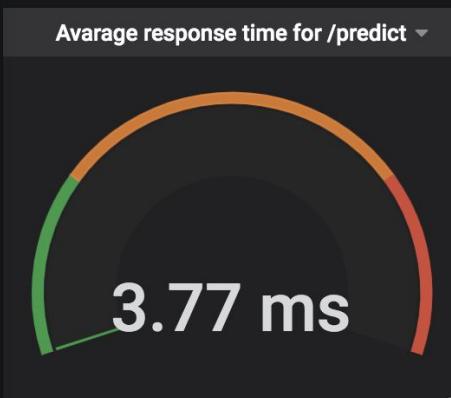
Implementation on production

- Python libraries, among others:
 - Flask
 - scikit-learn
 - Scipy, numpy
 - Joblib
 - Marshmallow
 - Webargs
 - Gunicorn
- This is not open-source for now :(



Monitoring model performance

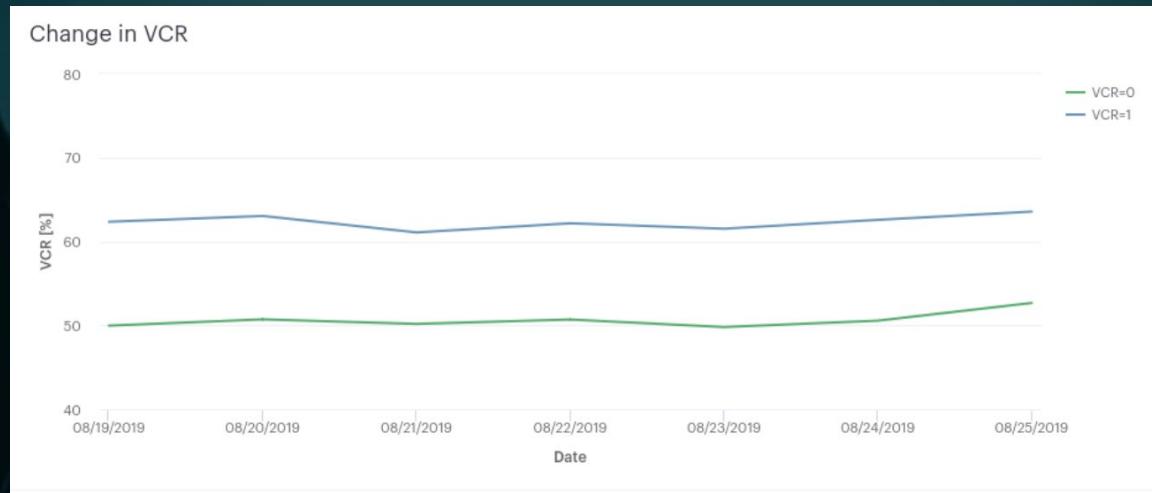
- Grafana health check of service



Example data - real traffic

Monitoring model performance

- Python:
 - Mode (SQL with Python support) - reporting daily on slack



Example data - real traffic

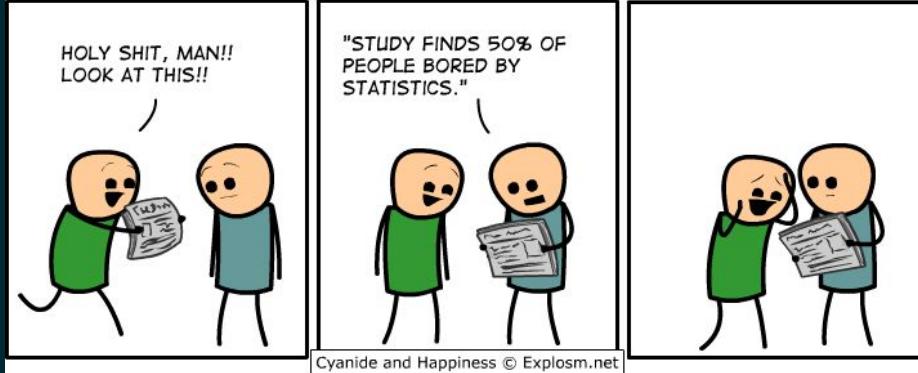
Monitoring model performance

We can observe that model is not changing its behaviour in time

- It's not worsening in performance with time
- We see actual difference in two predicted groups
- We can observe if groups does not decrease in size
- We can see how model performs when extending roll outs
- We can see that model is still working and predicting (aka health check of service)

Summary

- **Modelling projects are really for everyone and Python is a great programming language to do it with**
- **Github link to the presentation:**
 - https://github.com/martynaut/conferences_meetups/blob/master/pyconpl_130919/PyConPL2019_MUT.pdf



A vibrant illustration of Link from The Legend of Zelda: Ocarina of Time. He is shown from the waist up, wearing his signature red tunic, blue undershirt, and brown gauntlets. He holds his golden Hylian shield in his left hand, which features intricate designs of a sword and shield. In his right hand, he holds the hilt of his Master Sword. He has his characteristic long, pointed red hair and a determined, slightly grumpy expression. The background is a bright, cyan-colored gradient with diagonal motion streaks, giving it a dynamic feel.

Join us!

Current openings

<https://www.fandom.com/careers>

Android Developer

Poznań

Senior Data Engineer

Poznań, Poland

Front-end Developer

Poznań

Senior Dev Ops Engineer

San Francisco, CA

Full Stack Developer

Poznań

Senior Java/Microservices Engineer

Poznań, Poland



Go and be data
driven!

Questions?

If you have any feedback after this presentation, please send it
to [murbanek at fandom.com](mailto:murbanek@fandom.com)



Thank You

Martyna Urbanek-Trzeciak

Data Analyst

murbanek@fandom.com

github.com/martynaut