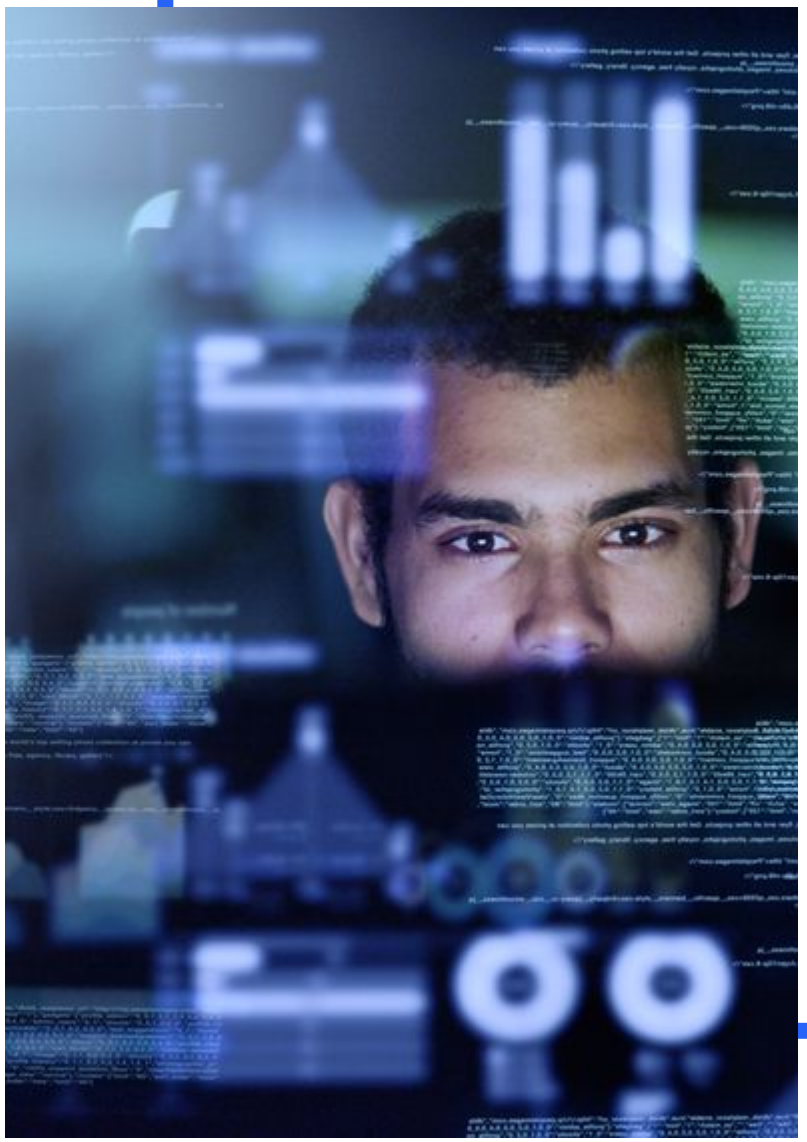
Flask vs Django -Два фреймворка для деплоя моделей

Преподаватель: Герард Костин

Проблемы продакшен для Keras

- Сериализация модели
  (PMML + DataFlow)
- Тяжелые библиотеки
  (лимит на закачку в AWS Lambda)
- Проблемы с обработкой батча



Flask and Django

Flask:

- Сервер разработки и отладки (2010 год)
- Простой
- Легко кастомизируемый
- Интегрированная поддержка модульного тестирования
- RESTful-совместимый
- Использует Jinja шаблонов
- Поддержка cookie (сеансы на стороне клиента)
- 100% WSGI 1.0 совместимый
- Unicode
- Обширная документация
- Совместимость с Google App Engine
- Доступны расширения для улучшения

Django:

- Легкий и автономный веб-сервер для разработки и тестирования
- Система сериализации и проверки HTML форм,
- ORM (object relation mapping)
- Кеширование
- Внутренняя система диспетчерезации
- Поддержка множества языков (автоперевод компонентов)
- XML и / или JSON
- Поддержка модульной тестовой среды Python
- Среда Django REST

Flask VS Django

Development speed

Простота Flask = быстрый запуск

Django был создан для быстрой разработки сложных веб-приложений. (легко масштабируемые, надежные и поддерживаемые веб-приложения в рекордно короткие сроки.)

Learning curve

Требует обширного документирования ввиду простоты и отсутствия стандартов

Django предоставляет преимущества, если вам нужно сменить команду в середине процесса разработки или масштабировать приложение уже после завершения проекта.

Flask VS Django

Гибкость и управляемость

Flask это минимализм и простота. Разработчик может реализовать все именно так, как он того хочет, используя огромный спектр внешних библиотек (гибкость и расширяемость)

Django, с его встроенными функциями и модулями, предлагает гораздо меньше свободы и контроля.

Community

Сообщество Flask в настоящее время не такое большое, и поэтому получить информацию
будет сложнее.

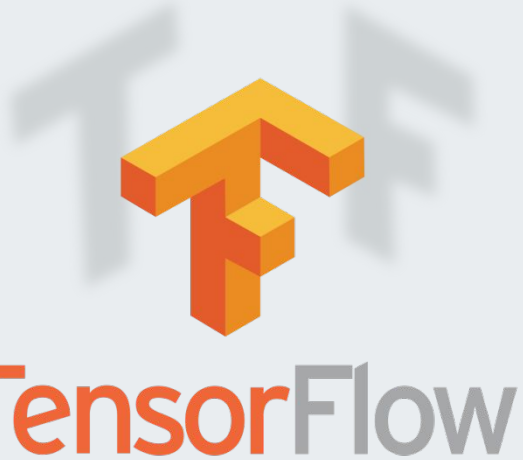
У Django огромное активное сообщество разработчиков (легко найти разработчиков для совместной доработки приложения, множество полезного контента, уже находящегося в свободном доступе).

Зрелость

Flask гораздо моложе, был представлен в 2010 году, поэтому у него не так много плагинов и расширений.

Django - это очень зрелая платформа, ее первый выпуск появился в 2005 году. Она собрала множество расширений, плагинов и сторонних приложений, охватывающих широкий спектр потребностей.

Flask + Keras



https://habr.com/ru/post/193242/

https://pymbook.readthedocs.io/en/latest/flask.html

- Werkzeug
  a WSGI utility library
- jinja2
  which is its template engine

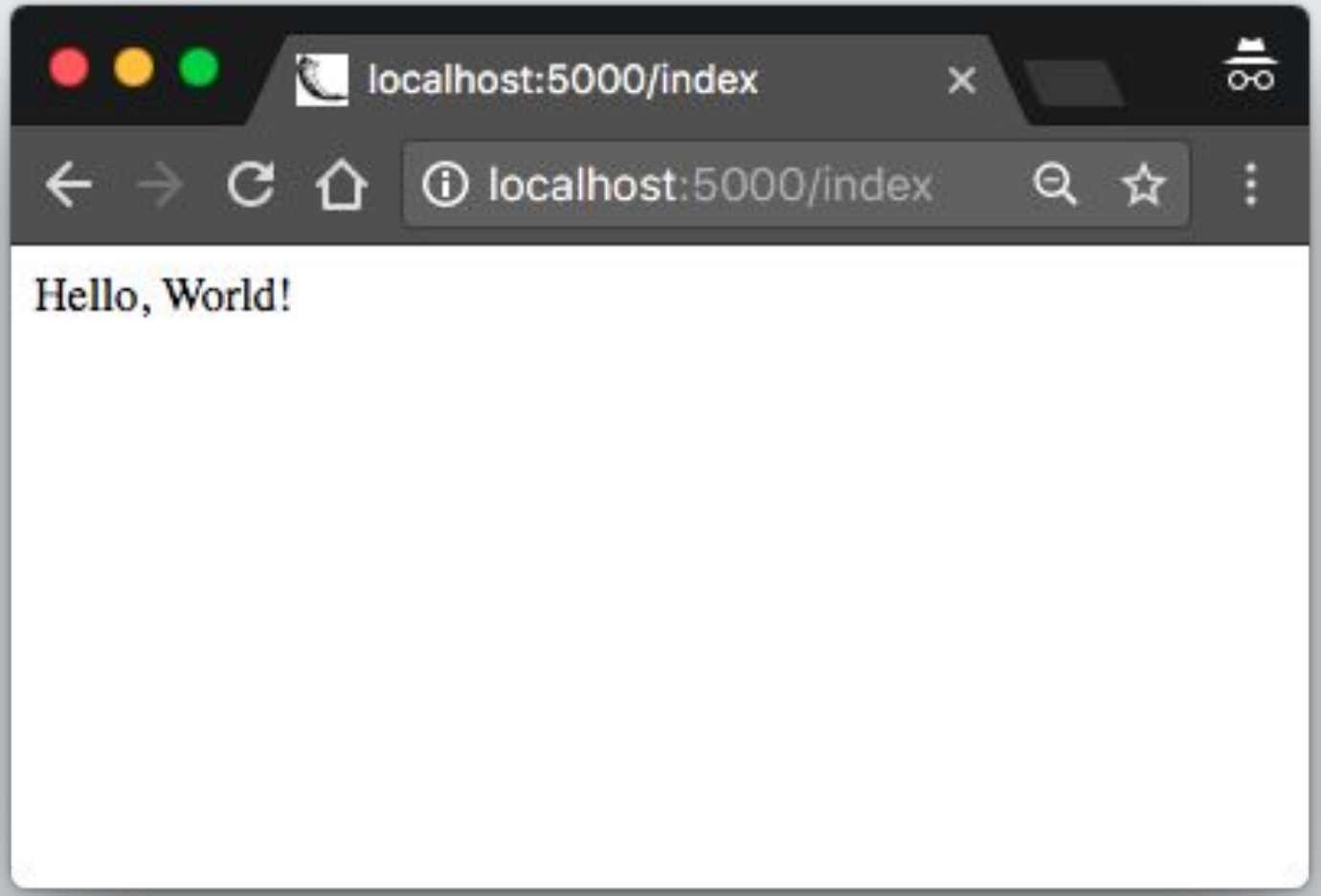
Flask CODE 1/2

```
import flask

 # Create the application.
 APP = flask.Flask(__name__)

@app.route('/')
@app.route('/index')
def index():
        return "Hello, World!"

if __name__ == '__main__':
        APP.debug=True APP.run()
```



https://pymbook.readthedocs.io/en/latest/flask.html

Flask CODE 2/2

```
# Загружаем билиотеки
import flask
import pandas as pd
import tensorflow as tf
import keras
from keras.models import load_model

# Инициализируем Flask
app = flask.Flask(__name__)


# Загрузка модели
global graph
graph = tf.get_default_graph()
model = load_model('games.h5')
```

```
# Задаем функцию Predict
@app.route("/predict", methods=["GET","POST"])
def predict():
    data = {"success": False}

    params = flask.request.json
    if (params == None):
        params = flask.request.args

    # if parameters are found, return a prediction
    if (params != None):
        x=pd.DataFrame.from_dict(params, orient='index').transpose()
    with graph.as_default():
        data["prediction"] = str(model.predict(x)[0][0])
        data["success"] = True

#Возвращаем результат json format
return flask.jsonify(data)

# Запускаем Сервер
app.run(host='0.0.0.0', port=5001)
```