

Building data products with Flask and AngularJS

Andy Goldschmidt

About me

- Data Scientist @ Jimdo --> **we're hiring!**
- ABBY.io
- Python for ~5 years

What's a data product?

An application that extracts value from data to improve and enhance the user's online experience.

- recommender systems (Amazon, Netflix)
- advertisements (retargeting, real-time bidding)
- predictions (customer relations, car electronics)

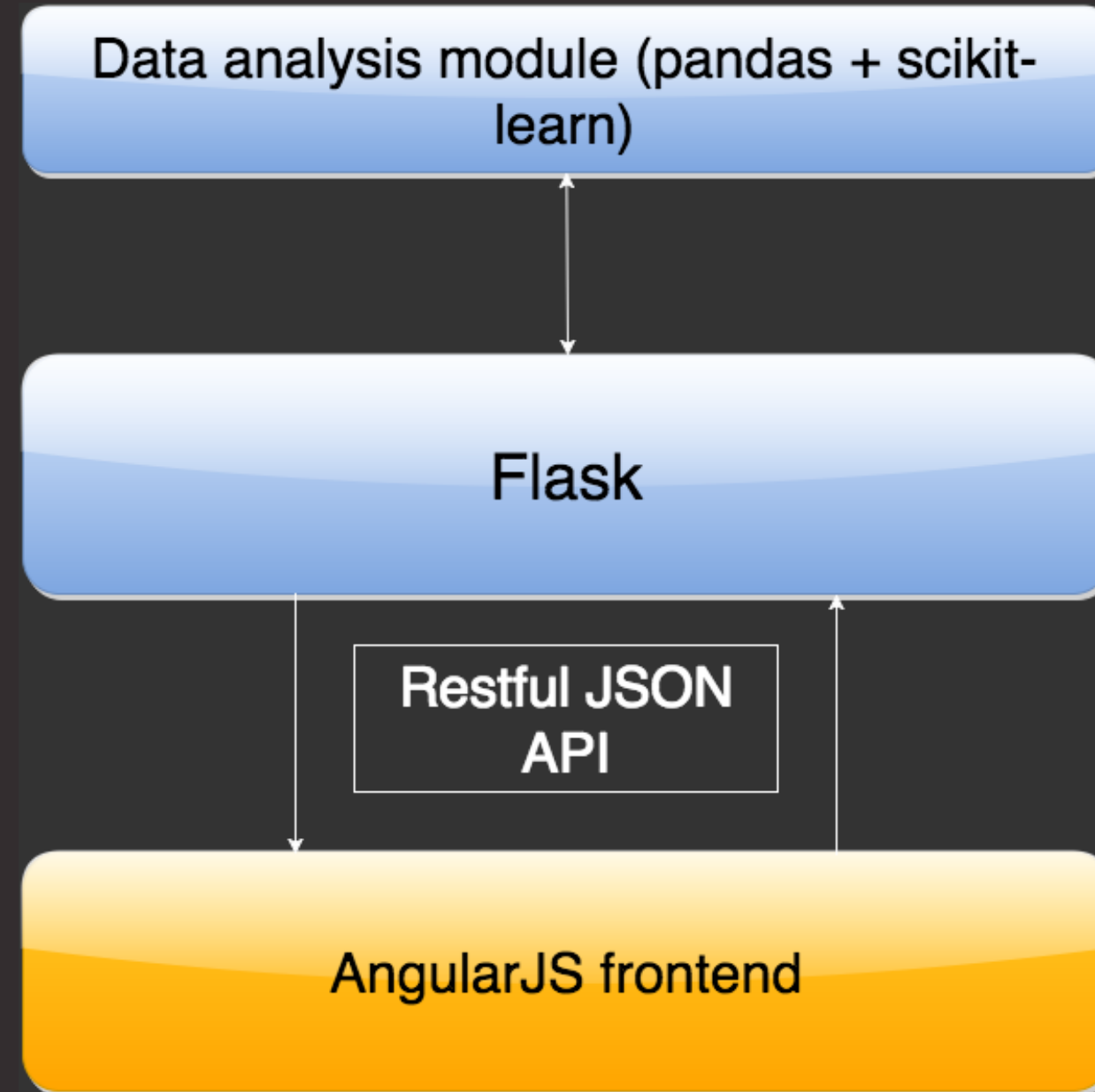
Our demo app

Actually, there are 2 apps:

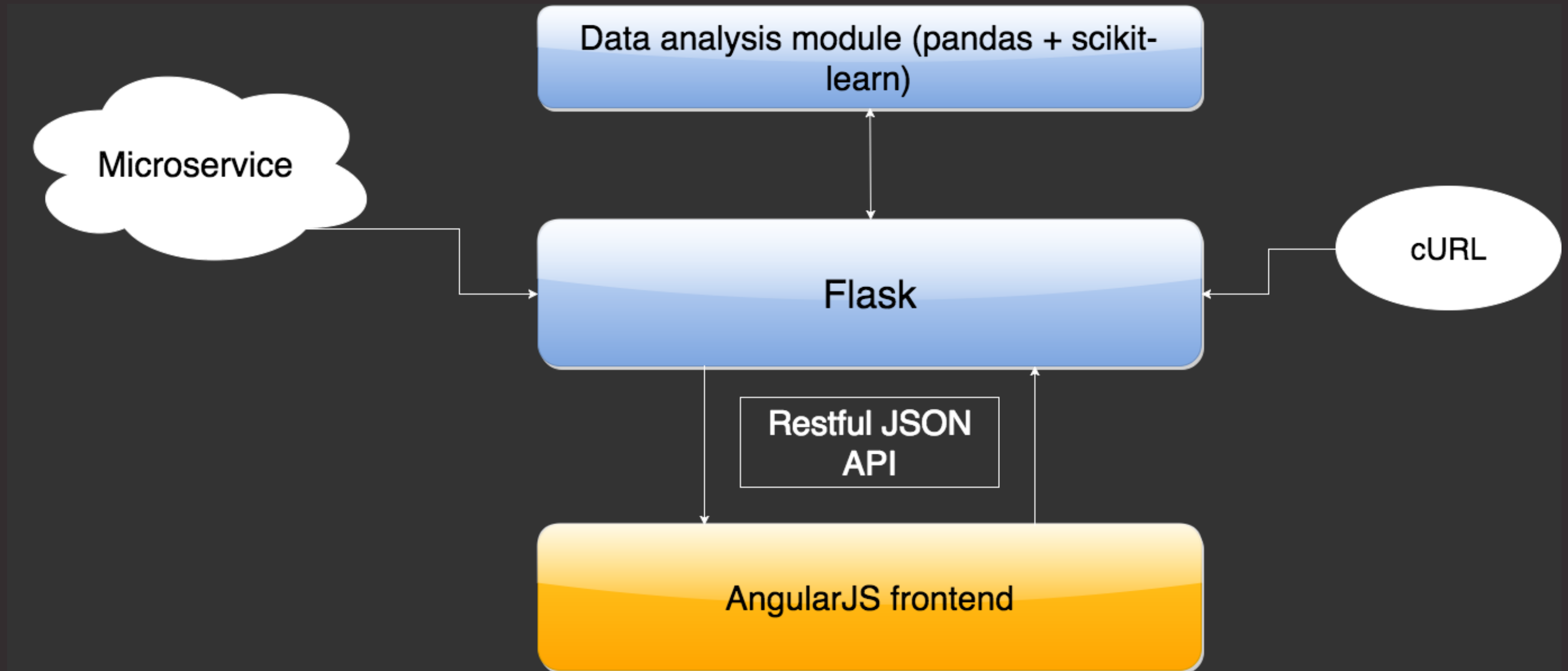
1. upload as CSV and classify your data
2. upload an image and get back dominant colors

Demo

Architecture



Architecture



Frontend

- **AngularJS** + angular-route + angular-file-upload
- Papaparse

AngularJS

HTML:

```
<body ng-app="app" ng-cloak>
  <nav class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#/">Demo</a>
      </div>
      <div id="navbar" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
          <li><a href="#/">Classifier</a></li>
          <li><a href="#/color">Color analyzer</a></li>
        </ul>
      </div>
    </div>
  </nav>

  <div class="container" style="padding-top: 50px">
    <div ng-view></div>
  </div>
  ...
</body>
```

AngularJS

Init & Config

```
var app = angular.module('app', ['ngRoute', 'angularFileUpload']);

app.config(['$routeProvider', function($routeProvider) {
    $routeProvider
        .when('/', {
            templateUrl: 'partials/classifier.html',
            controller: 'IrisController'
        })
        .when('/color', {
            templateUrl: 'partials/color.html',
            controller: 'ColorController'
        })
    });
});
```

AngularJS

Controller (presentation logic)

```
app.controller('IrisController', ['$scope', '$http', 'FileUploader', 'CsvParser',
function($scope, $http, FileUploader, CsvParser) {
    $scope.payload = {};
    $scope.classification = {};

    $scope.uploader = new FileUploader({
        url: 'http://localhost:5000/upload'
    });

    $scope.uploader.onAfterAddingFile = function(fileItem) {
        CsvParser(fileItem).then(function(res) {
            $scope.payload.data = res['data'];
            $scope.availableFields = res['fields'];
        });
    };

    $scope.send = function() {
        $scope.payload.features = parse($scope.featuresObj);
        $http.post('http://localhost:5000/classify', $scope.payload)
            .then(function(response) {
                $scope.classification.results = response['data']['results'];
                $scope.classification.labels = Object.keys(response['data']['results']);
                $scope.classification.accuracy = response['data']['accuracy'];
            }, function(err) {
                console.log(err);
            });
    };
}]);
```

Papaparse

CSV parsing (wrapped in Angular factory)

```
app.factory('CsvParser', function($q,$rootScope) {
    function parse(fileItem, callback) {
        Papa.parse(fileItem['_file'], {
            header: true,
            skipEmptyLines: true,
            worker: false,
            complete: function(results) {
                callback(results);
            }
        });
    }

    return function(fileItem) {
        var deferred = $q.defer();
        parse(fileItem, function(results) {
            var final = format(results);
            $rootScope.$apply(function() {
                deferred.resolve(final);
            });
        });
        return deferred.promise;
    };
});
```

Backend

- Flask
- scikit-learn + pandas

Flask

```
@app.route('/classify', methods=['POST'])
def classify_data():
    result = classify(request.json['data'],
                     request.json['features'],
                     request.json['targetColumn'])
    return jsonify({"results": result[0], 'accuracy': result[1]})

if __name__ == '__main__':
    if not os.path.isdir(app.config['UPLOAD_FOLDER']):
        os.mkdir(app.config['UPLOAD_FOLDER'])
    app.run(debug=True)
```

scikit-learn + pandas

```
def classify(data, features, target):  
    df = pd.DataFrame(data)  
    labels = df[target].unique()  
  
    train, test = split_data(df)  
    clf = train_model(train[features], train[target])  
    predictions = predict(test[features], clf)  
  
    cm = confusion_matrix(test[target], predictions)  
    accuracy = accuracy_score(test[target], predictions)  
  
    df_out = pd.DataFrame(cm, index=labels, columns=labels)  
    return df_out.astype(float).to_dict(), accuracy
```

Summary

- AngularJS:
 - flexible UI
 - asynchronous HTTP out-of-the-box
- Python:
 - Flask very lightweight
 - utilize power of machine learning libraries

The end

Check out the code: github.com/andygoldschmidt/iris-webapp

Twitter: [@datenheini](https://twitter.com/datenheini)