Maël FABIEN

Data Scientist - Actuary

(22 years old)

Nationality: France, Luxemburg

+(00)33 7 87 20 40 33

mael.fabien@gmail.com

7 rue Barrault, 75013 Paris

https://github.com/maelfabien

https://www.linkedin.com/in/mael-fabien/

http://maelfabien.fr

- French (Mother Tongue)
- English (C2)
- German (B2)

Interests

- Statistics
- Machine Learning
- Actuarial Science
- Time Series
- Quantitative Finance
- Artificial Intelligence
- Data Visualization
- Design/Web
- Sports

Stack

python, Apache Spark, Scala, Java, cassandra, mongoDB, scikit learn, PostgreSQL, TensorFlow, Keras

Education

BSc. Economics — *HEC Lausanne* — 2016

Statistics, Mathematics, Economics

MSc. Actuarial Science — *HEC Lausanne* — 2018

Risk modeling, Stochastic Processes

MS Big Data — *Telecom ParisTech* — 2019

Big Data Tools, Machine Learning, Deep Learning

Experience

pôle emploi — 09.2018 - 05.2019 : Affective Computing Research Project at Telecom Paristech

Research projet for the French employment center on multimodal **sentiment analysis**. Delivering state of the art models in Text, Sound and Video processing for sentiment analysis.

vaudoise — 02.2018 - 08.2018 : Actuary Intern in Non-Life department, Vaudoise Insurance

In charge of the **redesign** of a private Accident Insurance product. Market **analysis**, analysis of the current **portfolio** (structuring and extracting data (30K insured), statistical analysis, controlling for missing values, signaling incoherent manual entries), product **pricing** (MLE, **robustness** tests, profit testing, **simulations**).

Worked closely with the Subscription department. The pricing has been approved by the Actuarial Product Development Department. This internship lead to a **Master Thesis** and was awarded the **maximal grade** of 6/6.

Loomky — 2016 - 2018 : Entrepreneurial Projects

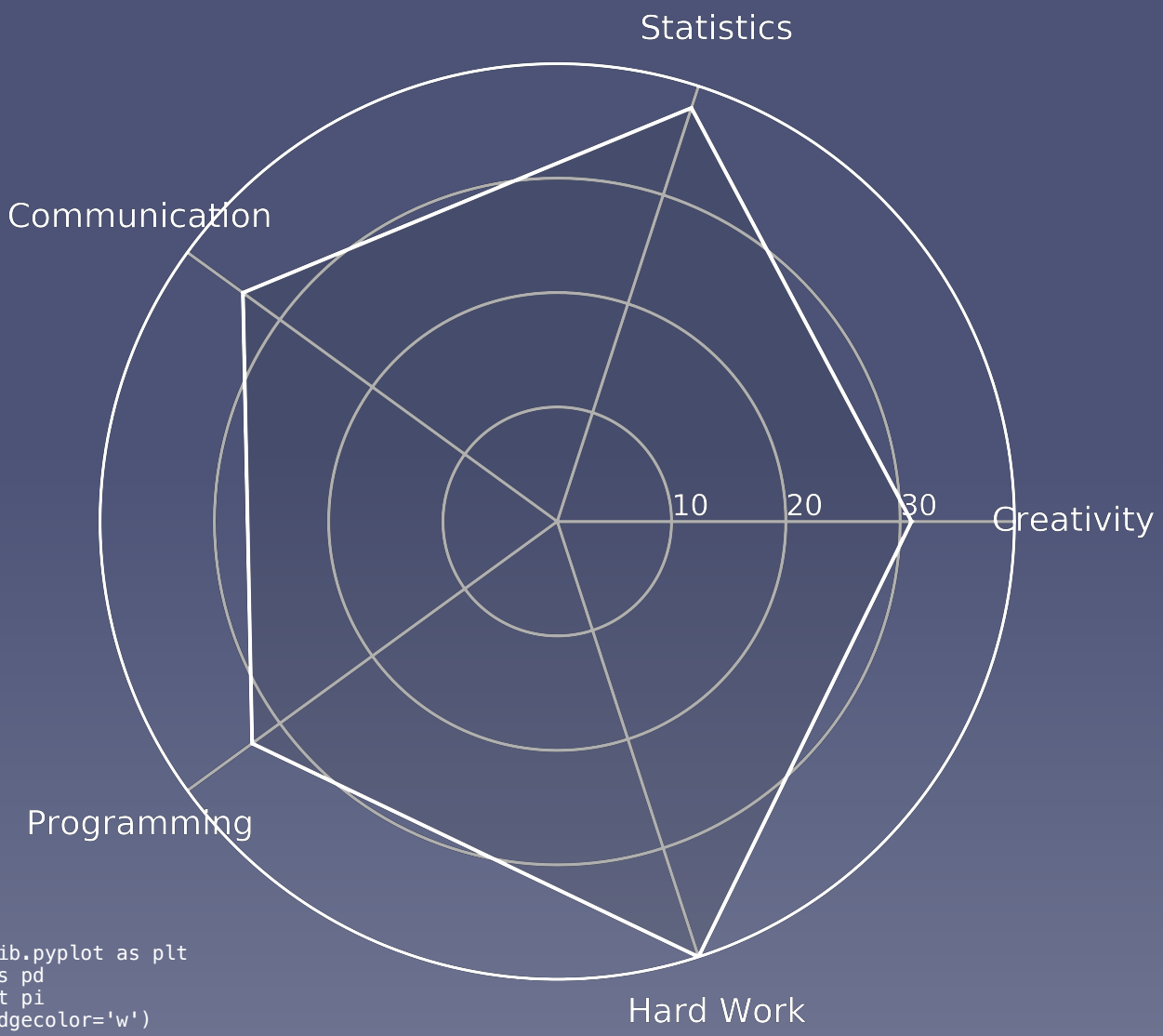
Jury Prize winner of the Entrepreneurial Contest Start Lausanne after a 6 months competition among 65 startups. I have worked 2 years on Wanago, a specialized crowdfunding platform, and I am now working on Loomky, a **digital rental solution** for car dealerships. **Test** phases on 8 dealerships. Partnerships with **Vaudoise** Insurance and BYmyCAR.

UNIL | Université de Lausanne HEC Lausanne — 09.2016 - 03.2018 : Teaching Assistant in Corporate and Public Finance

Assisted Professor Nikolov and then Professor Soguel for the 3rd year Bachelor course of **Corporate Finance and Public Finance.** This was a **40%** position on the side of my studies.

Projects

- A geolocation ML algorithm from a triangular signal strength (RSSI)
- A CNN for facial emotion recognition with Keras and Tensorflow
- A SVM classifier for facial landmark recognition with Opencv
- A CNN based autonomous driving car with Udacity simulator
- Predicting the next musical hit of 2019 (hackathon)
- Several Kaggle Competitions




```
# Libraries
import matplotlib.pyplot as plt
import pandas as pd
from math import pi
plt.rc('axes',edgecolor='w')

# Set data
df = pd.DataFrame({
'group': ['A','B','C','D'],
'Creativity': [31, 1.5, 30, 4],
'Statistics': [38, 10, 9, 34],
'Communication': [34, 39, 23, 24],
'Programming': [33, 31, 33, 14],
'Hard Work': [40, 15, 32, 14]
})

# number of variable
categories=list(df)[1:]
N = len(categories)

# We are going to plot the first line of the data frame.
# But we need to repeat the first value to close the circular graph:
values=df.loc[0].drop('group').values.flatten().tolist()
values += values[:1]
values

# What will be the angle of each axis in the plot? (we divide the plot / number of variable)
angles = [n / float(N) * 2 * pi for n in range(N)]
angles += angles[:1]

# Initialise the spider plot
ax = plt.subplot(111, polar=True)

# Draw one axe per variable + add labels labels yet
plt.xticks(angles[:-1], categories, color="white", size=8)

# Draw ylabels
ax.set_rlabel_position(0)
plt.yticks([10,20,30], ["10","20","30"], color="white", size=7)
plt.ylim(0,40)

# Plot data
ax.plot(angles, values, linewidth=1, linestyle='solid', color="white")

# Fill area
ax.fill(angles, values, 'w', alpha=0.1)

plt.savefig('radar.png', transparent=True, dpi=1000)
```