

Outputs des sprints de préparation du hackathon Eurostat de mars 2017

DMCSI - SGI

Organisation

Un projet de hackathon a été évoqué à l'occasion de la réunion de la Task Force Big Data en octobre 2016, les seules informations disponibles alors étaient :

- Le thème : "Skills"
- L'ambition : utiliser des technologies modernes (datascience, big data) et combiner des sources de données (traditionnelles, alternatives)
- 48 h par équipe de 3

Dans le but de préparer ce hackathon, nous avons formé une équipe pluridisciplinaire : Frédéric Comte, Romain Tailhurat, Laurent Bénichou, Eric Sigaud, Franck Cotton (SGI), Stéphanie Combes, Benjamin Sakarovitch (DMSCI) et Maxime Bergeat (DARES).




Préparation et follow-up

Le sujet précis du hackathon n'étant pas connu avant le début des festivités, les journées de préparation ont été inscrites dans la démarche commencée par la DARES d'évaluation des offres d'emploi disponibles sur Internet (en particulier le Bon Coin) pour la statistique publique.

Ces données peuvent être intéressantes pour **compléter des indicateurs** d'emplois vacants, de tension sur le marché de l'emploi, mais également pour l'**étude** plus micro des métiers, compétences recherchées, afin par exemple de mieux cibler les besoins du marché et de mieux orienter les chercheurs d'emploi en termes de formations professionnelles...

En particulier, à l'occasion des sprints : extraction, nettoyage, exploration et codification des offres en famille professionnelles.



Préparation et follow-up

5 sprints (journée de travail en équipe) thématiques :

- Décembre : scraping des données d'offre d'emploi du Bon Coin
- Janvier : datavisualisation des données (tests de différentes technologies)
- Février : tutoriel de machine learning en python
- Mars : préparation des données pour le hackathon
- Avril : discussion des suites et démonstration de Kibana par Louis Vignaud

Les différentes étapes sont décrites rapidement dans la suite.





Données scrapées : offres du Bon Coin

Données

59046 offres d'emplois ont été extraites du site le bon coin au milieu du mois de décembre 2016 en utilisant l'utilitaire (Scrapy).

date	-	23 décembre à 17:44
siren		412481301
desc	Nous cherchons le responsable financier & ...	
ref		1612cu
commune		Nice 06000
formation		Agent de maîtrise/Bac+3
sect		Services
exp		5 ans et plus
salaire		2 000 €
tps		Temps plein
contrat		Comptable unique (H/F)
offre		CDD
fonction		Comptabilité/Gestion/Finance

Description :

Assistants Managers (Futurs Managers) en Restauration Rapide H/F

Entreprise :

Rejoignez Eat Sushi, enseigne leader de la restauration rapide japonaise en livraison, vente à emporter et sur place. Eat Sushi dispose d'un réseau de plus de 20 restaurants en France.

Depuis plus de 10 ans, Eat Sushi affirme son positionnement haut gamme en proposant à la fois un service premium mais également des produits de haute qualité fabriqués sur place.

Dans le cadre de notre expansion, nous sommes à la recherche d'un Assistant Manager (Futurs Managers) H/F (75015) en CDI.

Data visualisation

Une fois scrapées, les données ont été représentées sous forme de datavisualisation pour permettre leur exploration multidimensionnelle et géographique (source de données supplémentaires : géolocalisation des communes par data.gouv.fr).

Technologie retenue pour la dataviz : Flask (python) pour créer l'appli, Bootstrap pour l'interface et dc.js (javascript) pour les graphiques.

Egalement testées lors des sprints : Bokeh (python), Rshiny, Kibana (Louis Vignaud)

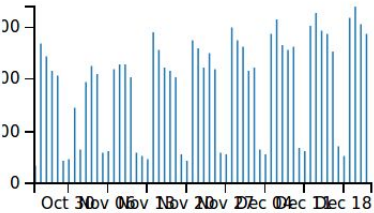
Le tableau de bord est complètement interactif.



Dataviz - Dashboard Flask-Dc.js-Bootstrap

Exploration des offres d'emploi du bon coin

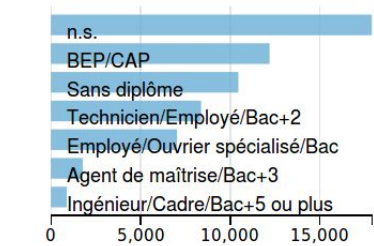
Nombre d'offres mises en ligne



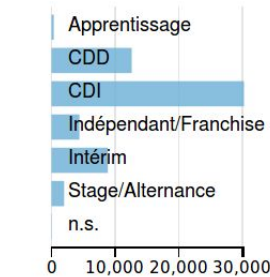
Secteur



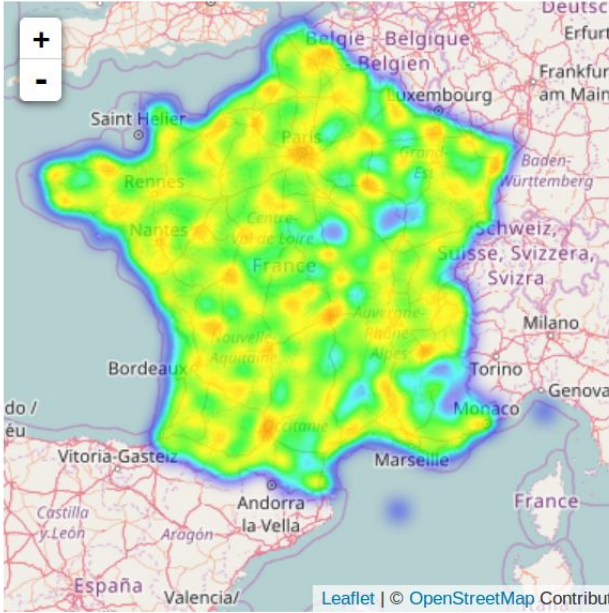
Formation



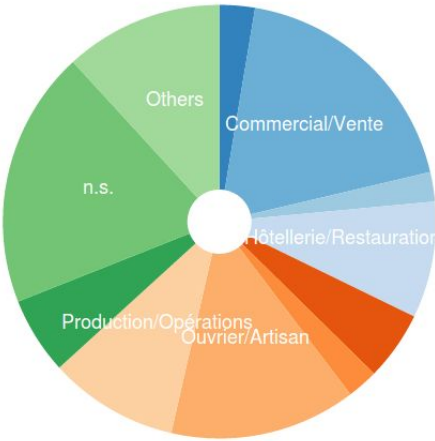
Contrat



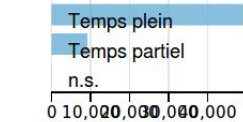
Communes d'origine



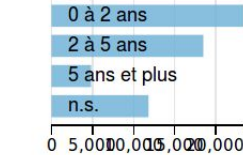
Fonctions



Temps de travail



Experience requise

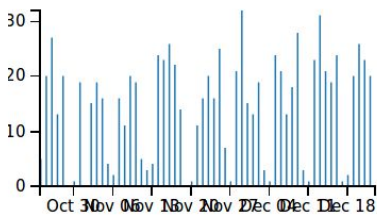


Nombre d'offres

58597

Exploration des offres d'emploi du bon coin

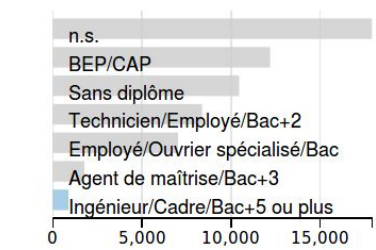
Nombre d'offres mises en ligne



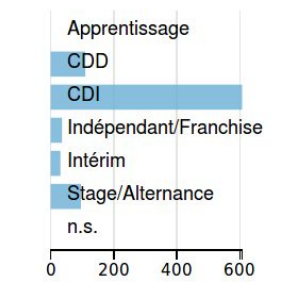
Secteur



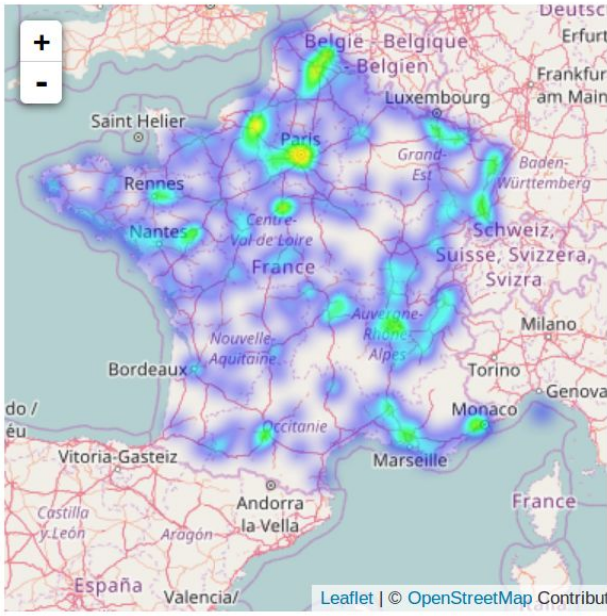
Formation



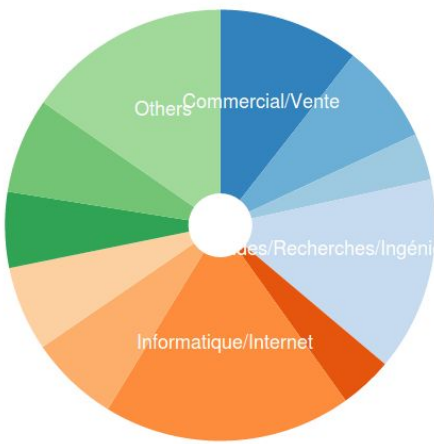
Contrat



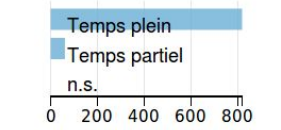
Communes d'origine



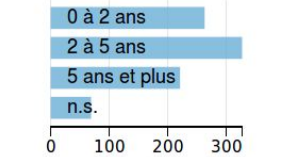
Fonctions



Temps de travail



Experience requise

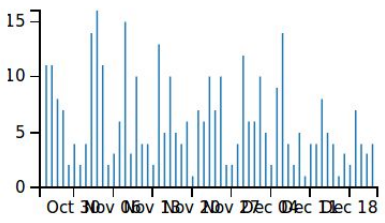


Nombre d'offres

880

Exploration des offres d'emploi du bon coin

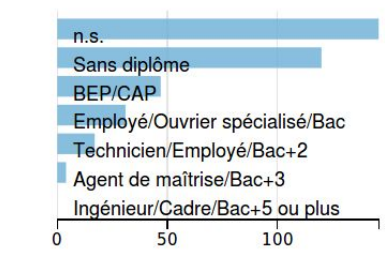
Nombre d'offres mises en ligne



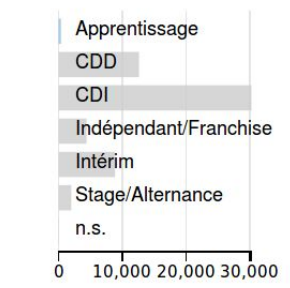
Secteur



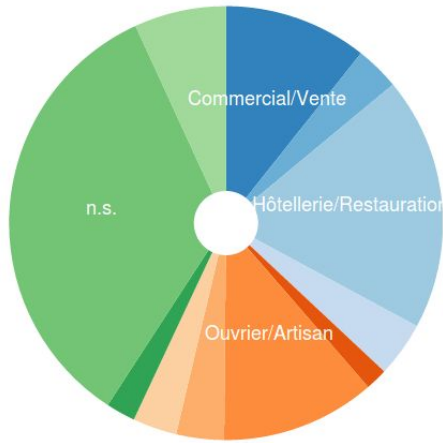
Formation



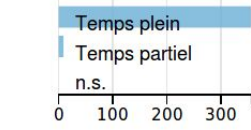
Contrat



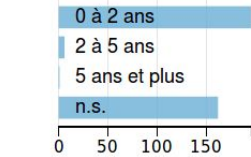
Fonctions



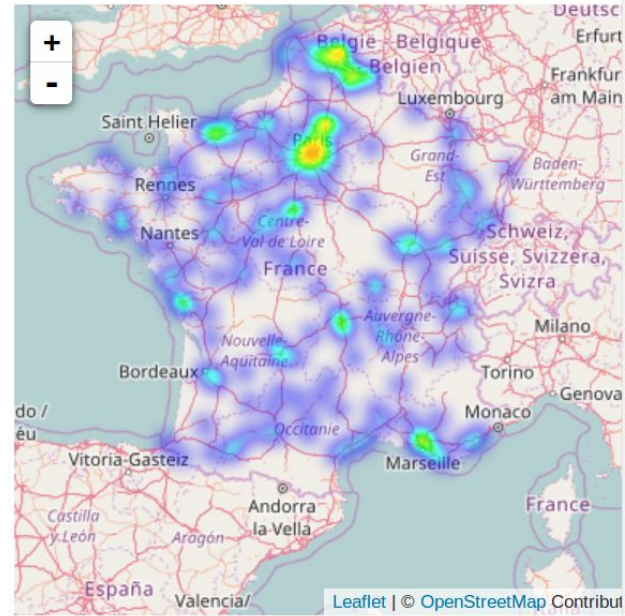
Temps de travail



Experience requise



Communes d'origine



Nombre d'offres

365

Dataviz - Dashboard Kibana



- Discover
- Visualize
- Dashboard
- Timeline
- Dev Tools
- Management

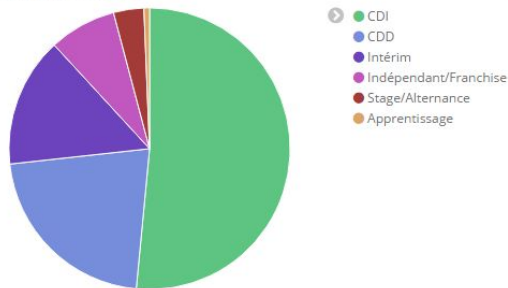
LeBonCoin > Salaire médian

57,458
Count

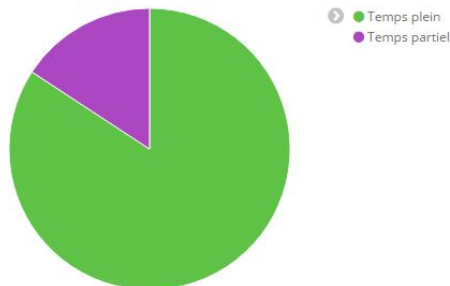
1,671.956

50th percentile of Salaire médian

LeBonCoin > Contrats



LeBonCoin > Temps Plein-Partiel

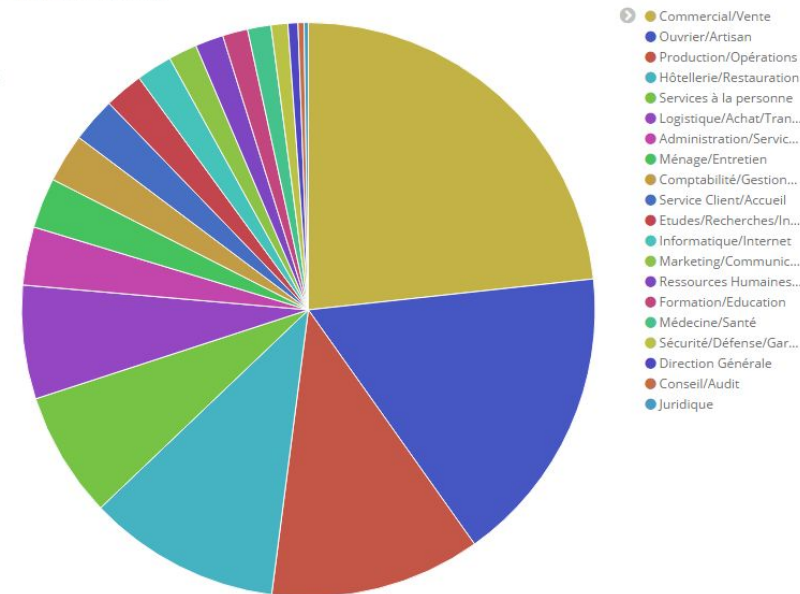


LeBonCoin > Desc_TagCloud

LeBonCoin > Salaire (1)



LeBonCoin > Fonction





- Discover
- Visualize
- Dashboard
- Timeline
- Dev Tools
- Management

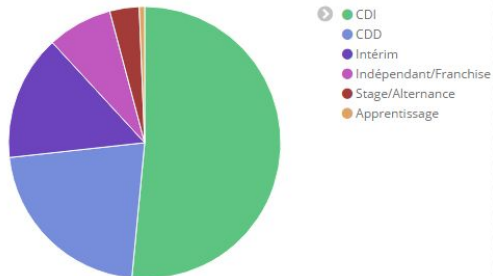
LeBonCoin > Salaire médian

57,458
Count

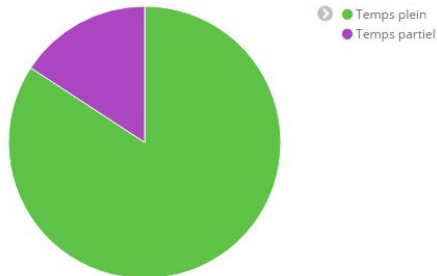
1,671.956

50th percentile of Salaire médian

LeBonCoin > Contrats

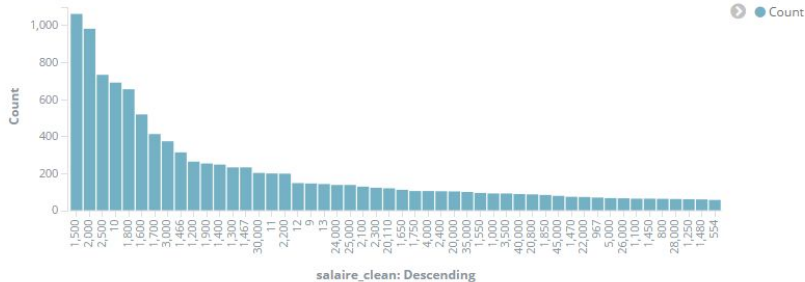


LeBonCoin > Temps Plein-Partiel

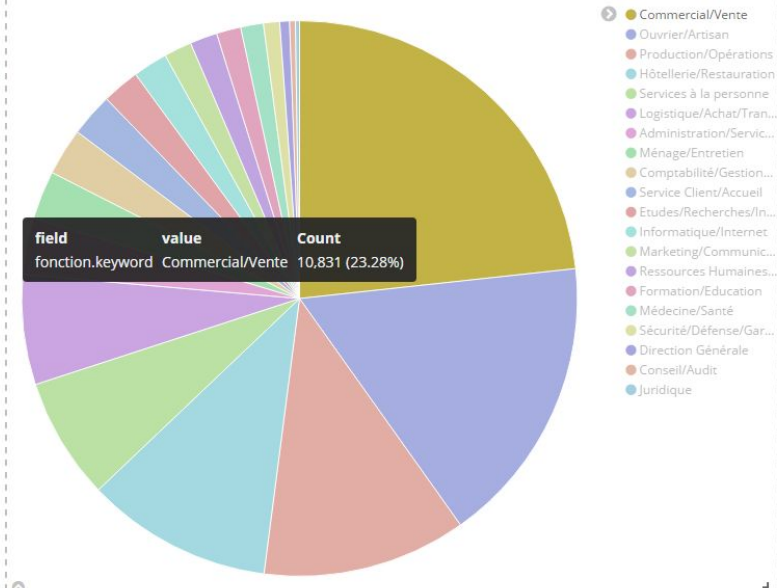


LeBonCoin > Desc_TagCloud

LeBonCoin > Salaire (1)



LeBonCoin > Fonction





Discover

Visualize

Dashboard

Timelion

Dev Tools

Management

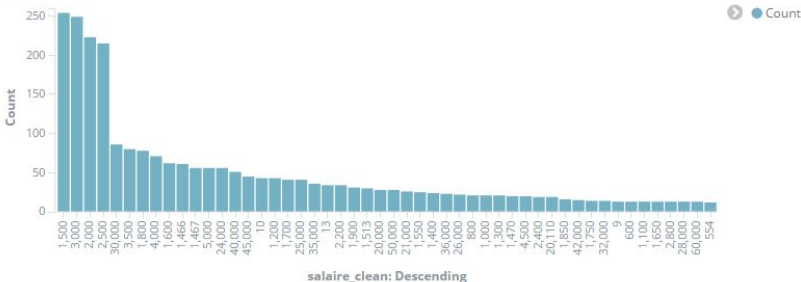
LeBonCoin > Salaire médian

10,831 2,000

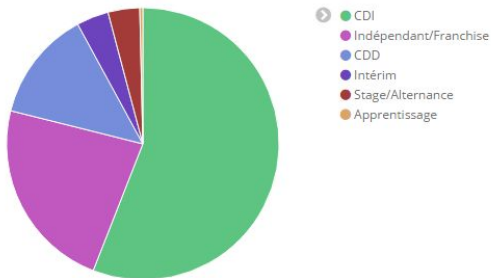
Count

50th percentile of Salaire médian

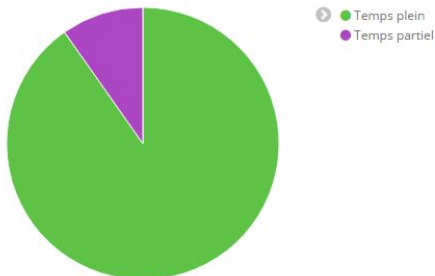
LeBonCoin > Salaire (1)



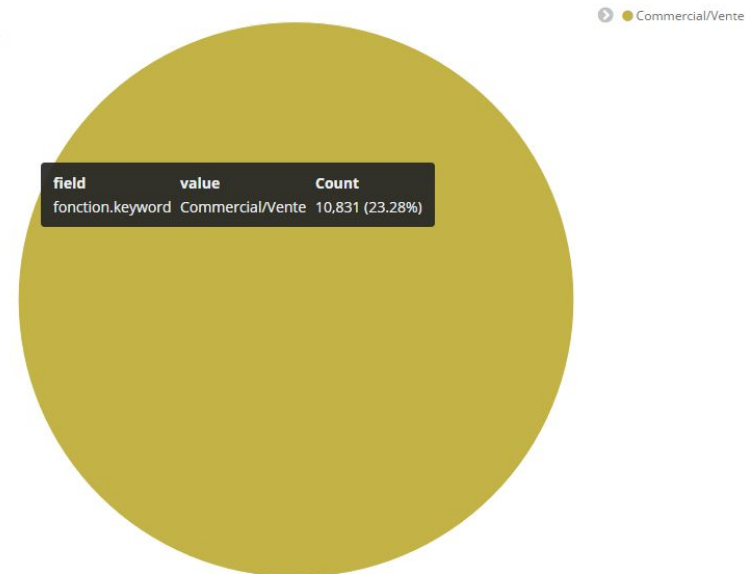
LeBonCoin > Contrats



LeBonCoin > Temps Plein-Partiel



LeBonCoin > Fonction



Collapse

LeBonCoin > Desc_TagCloud



Machine learning - codification

Codification des offres par machine learning

On dispose d'intitulés, on cherche des codes FAP (famille de métiers). Il existe des correspondances entre intitulés de métiers (environ 15000 dans la codification Rome) et FAP (table de passage Rome - FAP). On utilise cette correspondance pour étalonner un algorithme de classification supervisée.

De cette façon, nos intitulés peuvent se voir attribuer par l'algorithme "entraîné" un code FAP même s'ils ne sont pas formulés exactement de la même façon que les intitulés ROME, pourvu qu'ils intègrent des mots en commun ou des racines de mots en commun.

Technologie retenue pour la codification : scikit learn (python).



Codification des offres

Exemple de libellé Rome

libelle_rome	code_rome	fap2009
AGRICULTURE ET PÊCHE, ESPACES NATURELS ET ESPACES VERTS, SOINS AUX ANIMAUX	A	NA
Engins agricoles et forestiers	A11	NA
Opérateur / Opératrice d'épandage	A1101	A
Débardeur / Débardeuse	A1101	A
Conducteur / Conductrice de machines à vendanger	A1101	A
Conducteur / Conductrice d'abatteuses	A1101	A
Pilote de machines d'abattage	A1101	A
Conducteur / Conductrice d'engins forestiers	A1101	A
Conducteur / Conductrice de tracto-benne	A1101	A

offre

<fctr>

Intitulés des offres du Bon Coin

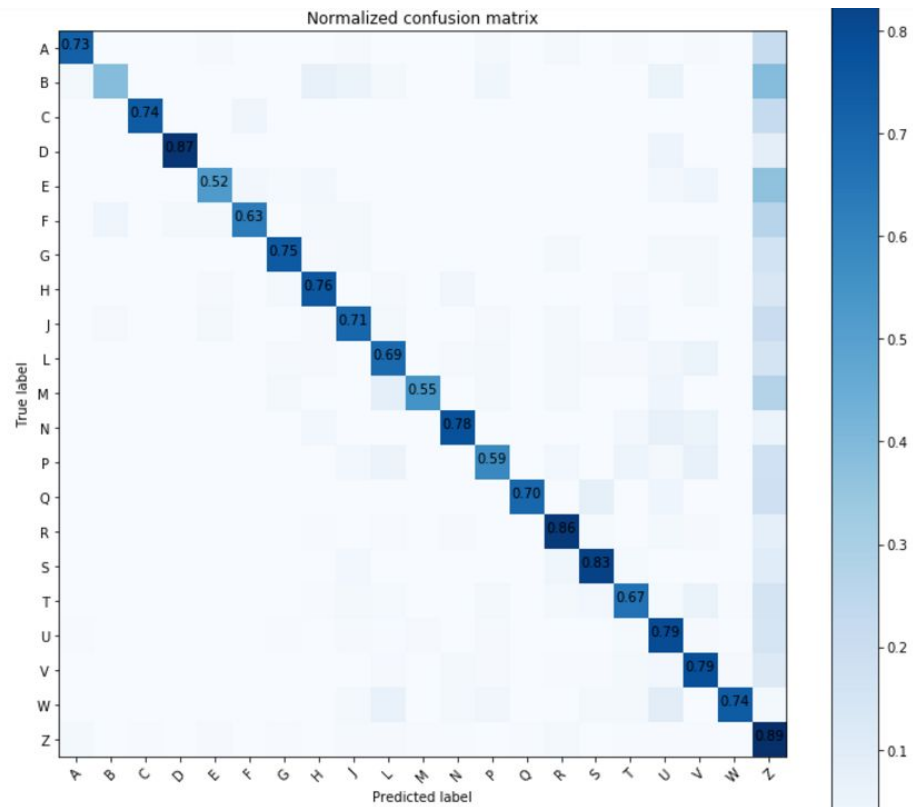
- 1 "Comptable unique (H/F)"
- 2 "URGENT Agent de propreté à Valdoie (H/F)"
- 3 "Vendeuse en Boutique de Mode - Saint-Brieuc (H/F)"
- 4 "Hôte / Hôtesse d'accueil (H/F)"
- 5 "Opératrice/opérateur sur machine numérique (H/F)"
- 6 "Conseiller/conseillère immobilier (H/F)"

Codification des offres

Ceci est rendu possible par les prétraitements appliqués aux différents champs textuels et leur transformation au format numérique par le biais de matrices documents x termes

	<i>call</i>	<i>time</i>	<i>date</i>	<i>conference</i>	<i>release</i>	<i>meeting</i>	<i>corporation</i>	<i>earnir.</i>
<i>document 1</i>	2	1	3	2	1	1	1	
<i>document 2</i>	1		2	1	2	1	1	1
<i>document 5</i>		1	2		2	1	1	1
<i>document 6</i>	1	2	1	1	3	1	1	1
<i>document 7</i>	1						1	
<i>document 8</i>			1		1		1	1
<i>document 9</i>	2		1	3	1	1	1	1
<i>document 10</i>	2	1		1	1		1	1
<i>document 13</i>					1			2
<i>document 14</i>							3	
<i>document 15</i>	1			2			1	2

Codification des offres

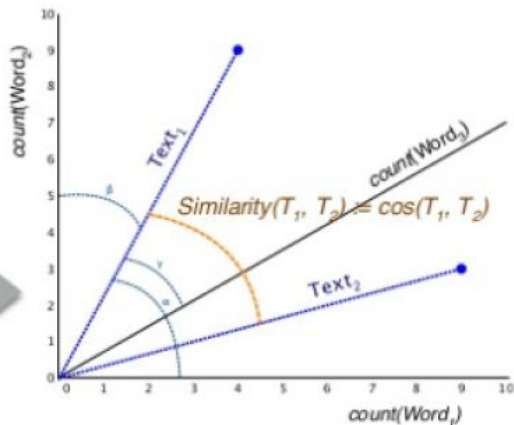


Codification des offres

Alternative testée : approche non supervisée, on compare directement les intitulés du bon coin avec les intitulés ROME via une métrique pertinente pour la comparaison de chaîne de caractère (la similarité cosinus).

Text 1: He that not wills to the end neither wills to the means.
Text 2: If the mountain will not go to Moses, then Moses must go to the mountain.

tokens	Text 1	Text 2
end	1	0
go	0	2
he	1	0
if	0	1
means	1	0
Moses	0	2
mountain	0	2
must	0	1
not	1	1
that	1	0
the	2	2



$$sim(\vec{x}, \vec{y}) = cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| |\vec{y}|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \sqrt{\sum y_i^2}}$$

Codification des offres

Les résultats peuvent être satisfaisants.

offre <fctr>	libelle_rome <chr>
1 "Comptable unique (H/F)"	Comptable unique
2 "URGENT Agent de propreté à Valdoie (H/F)"	Agent / Agente de propreté hospitalière
3 "Vendeuse en Boutique de Mode - Saint-Brieuc (H/F)"	Responsable de boutique
4 "Hôte / Hôtesse d'accueil (H/F)"	Hôte / Hôtesse d'accueil
5 "Opératrice/opérateur sur machine numérique (H/F)"	Opérateur / Opératrice sur machines numériques de reprographie
6 "Conseiller/conseillère immobilier (H/F)"	Conseiller / Conseillère agricole



Moi aussi je veux le faire

Codes

Sur le github <https://github.com/SSP-Lab/sprints-ntts-hackathon-2017> sont déposés :

- Les codes pour la datavisualisation avec Flask-dc.js-bootstrap
- Les codes pour la datavisualisation avec Kibana
- Le code pour l'exercice de machine learning de codification des offres
- Le code du scraping n'est pas rendu publique mais peut être demandé à l'équipe ainsi qu'une documentation de Scrapy.

Le prototype proposé à l'occasion du hackathon est présenté dans une présentation distincte.



