Stage Ingénieur de fin d'étude

Stage réalisé par
Simon Bussy

Au sein du groupe
« Plateform et Middleware »

Chez
Orange Silicon Valley

Sous la tutelle de
M. Garg Shishir et M. Quintuna Xavier - OSV
M. Jakubowicz Jérémie - Télécom SudParis

Du 17/02/2014 au 22/08/2014

Sommaire

"Data is a precious thing and will last longer than the systems themselves."

Tim Berners-Lee,
Inventeur du World Wide Web.

A mes parents,
A mes sœurs,
A ma copine

Remerciements

Tout d'abord, je tiens à remercier l'ensemble du groupe «*Platform and Middleware*», ainsi que l'ensemble de l'équipe d'Orange Silicon Valley pour leurs aides, leurs précieux conseils et tous les agréables moments passés auprès d'eux. Je remercie également l'ensemble des stagiaires sans qui la découverte de la Silicon Valley n'aurait pas eu la même saveur.

Je tiens à remercier tout particulièrement Shishir Garg qui m'a accordé sa confiance pour intégrer son équipe, mais également Xavier Quintuna qui a été d'une aide précieuse et qui a œuvré au bon déroulement de mon stage.

Je souhaite également remercier Jérémie Jakubowicz, mon tuteur à Télécom SudParis, qui a toujours été à l'écoute et m'a procuré de bons conseils, comme à son habitude. Je remercie aussi le Président du jury de bien vouloir lire ce rapport et m'évaluer.

Et enfin, je remercie mes parents pour leur soutien sans qui je n'aurais jamais pu connaître et vivre une telle expérience.

Résumé de la mission

Au cours de ce stage, j'ai étais plongé dans un univers dont rêverait tout passionné de nouvelles technologies. Et plus particulièrement en ce qui me concerne, être au cœur des rescentes perçées dans le domaine de l'intelligence artificielle a été une formidable expérience que je suis quelque peu attristé de devoir déjà achever.

Mon projet premier a été celui de créer un système capable de proposer des prédictions concernant l'évolution des tendances au sens large ; qu'il s'agisse de concepts, de technologies ou de start-ups. Et pour cela, l'idée est d'écouter ce qui se passe sur la toile et d'entraîner des modèles mathématiques à affiner peu à peu leurs prévisions, le tout en utilisant les dernières technologies visant à paralléliser les calculs sur un ensemble de machines distinctes appelé *cluster* et permettant de traîter de grosses quantités de données à la fois.

Pendant ces six mois, une autre de mes missions a aussi été, à l'image de celle d'un journaliste, de srcuter les mouvements au sein de la sphère *Big Data,* pour ensuite les étudier et les comprendre. J'ai alors eu le privilège d'être en interaction avec de nombreuses entreprises en lien, de près ou de loin, avec le traitement des quantités gigantesques de données.

Ces données, nous les produisons chaque seconde pour accroître ce formidable terrain de jeu qu'elles représentent, reflétant les connaissances et les comportements humains, et elles s'amoncellent de plus en plus vite dans un réceptacle infini grâce à internet.

Introduction

Aujourd'hui, une entreprise telle que le groupe Orange se doit d'innover continuellement pour rester compétitive et offrir de nouveaux services à ses clients. C'est pourquoi Orange a implanté un centre d'innovation du nom d'Orange Silicon Valley (OSV) situé dans la capitale de l'innovation mondiale : la Silicon Valley.

En étant présent dans cette région extrêmement dynamique, Orange a la volonté de rester à la pointe de la technologie et ainsi de lier des partenariats intéressants au service de ses clients.

J'ai eu la chance de pouvoir effectuer mon stage de fin d'études au sein de cette entité et de découvrir un monde de technologies tourné vers l'avenir et l'innovation.

Dans ce rapport, je présente justement dans une première partie l'environnement de mon stage, pour ensuite décrire plus précisément en seconde partie ma mission à Orange Silicon Valley. La dernière partie concerne les difficultées que j'ai pu rencontrer pendant cette période, mais aussi tout ce qu'elle m'a apporté, avec un avis personnel donné sur différents points.

Partie 1 :

L'environnement de la mission

1.1 Présentation de l'entreprise

FRANCE TELECOM est l'un des leaders mondiaux dans le domaine des télécommunications. Le groupe, qui commercialise la majorité de ses services sous la marque Orange, sert 236 millions de clients sur les 5 continents et compte 165000 employés dans le monde.

Le chiffre d'affaires consolidé du groupe pour l'année 2013 s'élève à 40.9 milliards d'euros. Le groupe a 3 domaines d'activités principaux : les télécommunications dans le foyer (téléphonie fixe, internet), les télécommunications mobiles et les services aux entreprises.

FRANCE TELECOM emploie plus de 167 millions de collaborateurs et la structure du groupe est très complexe.

Pour simplifier les choses, le groupe est divisé en six sous-groupes travaillant sur ses trois domaines d'activités :

- Home Group : centré sur le foyer, travaille notamment sur la téléphonie fixe mais aussi la domotique, et le M2M c'est-à-dire la communication de machine à machine.
- Personal Group : centré sur les services mobiles pour les particuliers.
- Enterprise : tourné vers les entreprises.
- Audience and Advertising Group : centré sur la monétisation des contenus vidéo ou internet.
- Content : tourné vers la stratégie d'acquisitions et de distribution de contenus.
- Healthcare : tourné spécifiquement vers les entreprises de la santé.

Les trois derniers groupes font partie des nouvelles activités de croissance de FRANCE TELECOM.

1.2 France Telecom et la recherche

France Telecom est l'un des leaders mondiaux dans le domaine de l'innovation et en tant que tel, dispose d'un réseau de chercheurs, de laboratoires et de techno-centres travaillant dans le domaine de la recherche et de l'innovation. Le centre de San Francisco, créé en 1999 sous le nom d'Orange Labs, avait pour but d'identifier de nouvelles technologies et de développer des prototypes dans le cadre de la recherche.

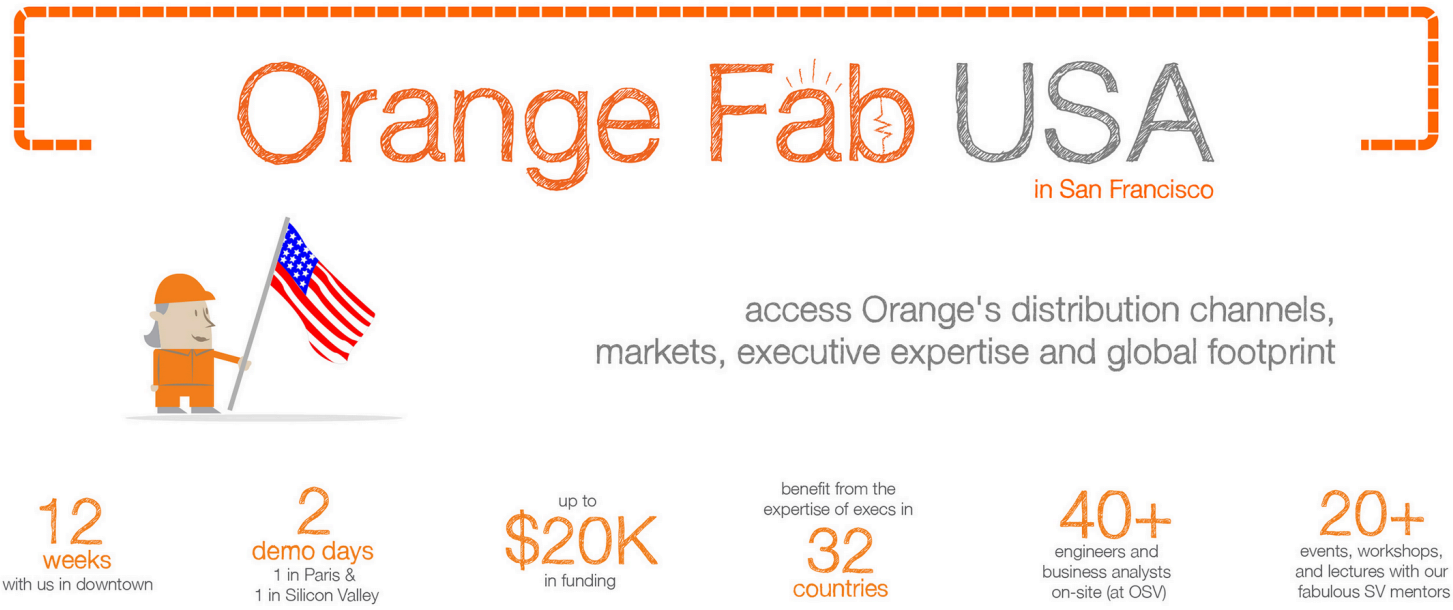
Les activités de recherches s'inscrivent sur le long terme et doivent apporter un avantage concurrentiel : l'idée est de constamment rester à la pointe de tout ce qui se fait dans le monde des hautes technologies. Cependant en mars 2010, la nouvelle équipe de management a décidé de faire passer le centre de San Francisco de la direction de la recherche et du développement à la direction des partenariats et des initiatives stratégiques.

L'avantage du centre de San Francisco est avant tout sa localisation : la Silicon Valley, au cœur d'un des centres majeurs de l'innovation mondiale. Toutes les entreprises High Tech y ont soit un siège, soit au moins une antenne : Google, Facebook, Microsoft, Apple ou encore Oracle pour n'en citer que quelques uns. Sans oublier bien sûr le nombre de start-ups créées ici chaque année, les fonds levés dans le cadre de Venture Capitalism pour leurs financements, et également le monde académique représentés entre autres par les universités de Stanford et Berkeley.

On comprend alors pourquoi la Californie est l'un des états des Etats-Unis contribuant le plus aux performances du pays en termes de Prix Nobel remportés. Internet a été inventé dans la Silicon Valley, et c'est ici que les idées fleurissent pour déterminer selon quelles modalités les êtres humains communiqueront, se déplaceront, et même plus généralement vivront dans les années futures. Autant de raisons qui justifient la présence essentielle d'un groupe de télécommunications comme FRANCE TELECOM à San Francisco, malgré le fait que le groupe n'a aucune activité économique sur le territoire américain.

Aujourd'hui le travail mené par le centre de San Francisco est avant tout un travail de veille stratégique puis de mise en action des conclusions de cette veille : connaître les nouvelles tendances et les impacts de ces modes de communication, identifier les entreprises innovantes à suivre dans ce domaine et créer des partenariats avec elles, pour enfin développer des prototypes à destination de la France. Le but est de rester à la pointe de la technologie dans tous les domaines de l'innovation et d'illustrer comment FRANCE TELECOM peut se positionner pour tirer le meilleur parti de ces nouvelles tendances identifiées.

Une entité un peu particulière est également présente à OSV : Orange Fab. Il s'agit un programme d'accélération de start-ups qui célébrait cette année sa troisième saison. Son rôle est de sélectionner plusieurs start-ups, pour ensuite les soutenir et les aiguiller dans leurs projets.



1.3 L'équipe et les bureaux de San Francisco

L'équipe d'Orange Silicon Valley est composée d'une soixantaine d'experts d'une vingtaine de nationalités différentes repartis par projets. Beaucoup de choses ont changé en terme d'organisation interne entre le début de mon stage et sa fin. Lors des premiers mois, nous étions repartis par équipes, chacune ayant un manager technique et un rôle bien précis. Les équipes représentaient plus ou moins la division de l'entreprise : les groupes *home*, *personal, consumer, enterprise* ainsi que les groupes techniques transversaux *access & core networks* et *platform & middleware* dans lequel j'ai été intégré.

Ces différentes équipes et managers étaient supervisés par le vice-président et le président d'*Orange Silicon Valley*. En juin 2014, l'organisation interne a complètement changé, passant d'un travail par équipe vers un travail par projet. La modification majeure a été le départ des directeurs techniques qui géraient ces équipes, les projets étant maintenant pris en charge par les employés eux mêmes et approuvés directement par le vice-président. Dans la dernière partie de ce rapport, je reviendrai sur mon ressenti personnel face à cette expérience peu banale qu'a été cette réorganisation.

Tous les postes de travail sont situés dans la même pièce : l'*open-space*. Seuls le CEO du centre, le VP (Vice Président) et la directrice des ressources humaines disposent d'un bureau personnel. L'ambiance de travail est à la fois studieuse et détendue : pas de code vestimentaire, des relations simples et non hiérarchiques entre collègues.
Un grand calme règne à notre étage, probablement justement parce qu'étant sur un espace ouvert, tout bruit devient vite pénible pour les autres collaborateurs. Au sein de mon groupe, nous communiquons via messagerie instantanée, alors même que nous sommes assis les uns en face des autres. Ceci nous permet de nous échanger en toute discrétion des liens vers des articles intéressants pour les sujets sur lesquels nous travaillons, de discuter ponctuellement de nos projets ou de se mettre d'accord sur des rendez-vous.

Et si besoin, nous convenons de nous déplacer vers une des nombreuses salles de réunion ou vers la cuisine, qui est un espace commun de détente. Le café gratuit, comme toutes les boissons, y est alors religion comme dans tout le reste de la Silicon Valley.

Partie 2 :

Ma Mission

2.1 Mes objectifs principaux

Si j'ai été accepté en stage chez Orange Silicon Valley, c'est pour mes connaissances en *Machine Learning* qui est un domaine de l'intelligence artificielle, situé entre les mathématiques et l'informatique. Mon projet de fin d'étude sur le *Deep Learning* et mes connaissances dans ce nouveau domaine en pleine émergence ont également énormément plu. Une de mes toutes premières tâches a d'ailleurs été celle d'expliquer plus précisément à une partie de mes collègues ingénieurs ce qu'était le *Deep Learning* et quelles promesses et applications pouvait-on attendre de ce nouveau *buzz word*.

Je vais donc naturellement commencer par présenter de manière générale en quoi consiste ce nouveau domaine.

Le *Deep Learning* est avant tout un sous-domaine du *Machine Learning* qui est une branche de l'intelligence artificielle. Le *Machine Learning* est la science visant à créer des algorithmes capables d'apprendre des comportements, ou concepts, à partir d'exemples qu'ils observent ; un peu comme un enfant apprend de son environnement lorsqu'il grandit.

Voici ci-dessous quelques exemples récents d'applications directes du *Machine Learning* :

Siri : Il s'agit là de la reconnaissance de la parole, où l'algorithme s'adapte à l'accent et la façon de parler du locuteur, et améliore la précision de la compréhension au fil du temps.

Le thermostat Nest : Celui-ci apprend les préférences de température de son possesseur, et ce suivant les différents moments de la journée. Il baisse alors intelligemment la température lorsque personne n'est dans la maison pour ne pas gaspiller d'énergie.

La Google car : Il s'agit de la voiture autonome, soit sans conducteur humain, développée par Google. Celle-ci observe son environnement et réagit en temps réel à des obstacles inattendus survenant sur la route, comme des piétons.

Le *Deep Learning* constitue une approche d'analyse de données basée sur l'empilement successif de réseaux de neurones artificiels (ou *ANN : Artificial Neural Network*, en anglais) qui peuvent, par une phase d'apprentissage, tenter de comprendre des phénomènes complexes.

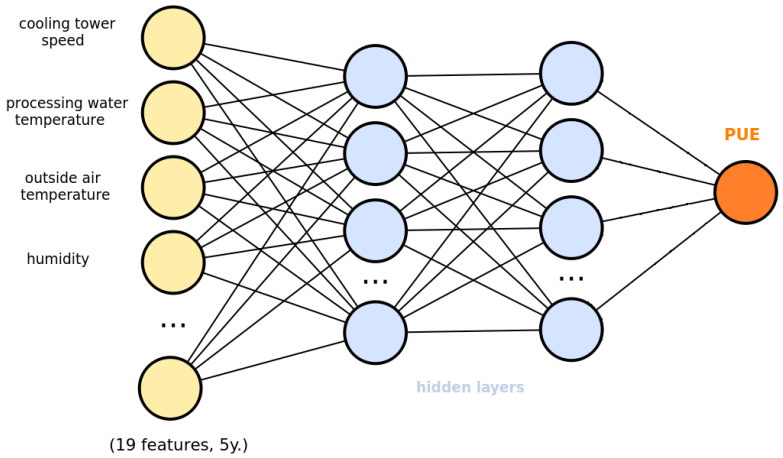
Il est donc nécessaire de s'arrêter un instant sur les réseaux de neurones. Il s'agit là d'un algorithme de *Machine Learning* qui est né dans les années 40, alors que des avancées sérieuses ont eu lieu dans le domaine des neurosciences, décrivant notamment comment fonctionnent les neurones d'un cerveau et comment est possible cette tâche si complexe qu'est l'apprentissage.

Sans rentrer ici dans les détails techniques, l'idée de base des réseaux de neurones est d'imiter le fonctionnement du cerveau, ce qui est évidemment un challenge fou. En effet le cerveau peut apprendre à voir, il traite alors des images ; mais aussi à entendre, il traite alors des signaux sonores ; et une immensité d'autres facultés qui nous sont par conséquent offertes, comme faire des mathématiques. Et il apprend à faire tout cela, non pas avec des centaines de différents programmes, mais bel et bien avec un seul et unique « algorithme d'apprentissage ». Il existe en effet aujourd'hui des preuves de cela, qui ont été initiées par l'expérience suivante.

Le cerveau interprète les sons ainsi : les oreilles captent le signal sonore qui est ensuite routé jusqu'au cortex auditif. En coupant alors le canal reliant les oreilles au cortex auditif, et en « connectant » les yeux au cortex auditif via le nerf optique, le cortex auditif va alors être capable d'apprendre à voir. Ainsi, la même zone du cerveau peut traiter différents types de données. Le cerveau apprend donc de lui même comment traiter les données, et ce via un unique algorithme d'apprentissage.

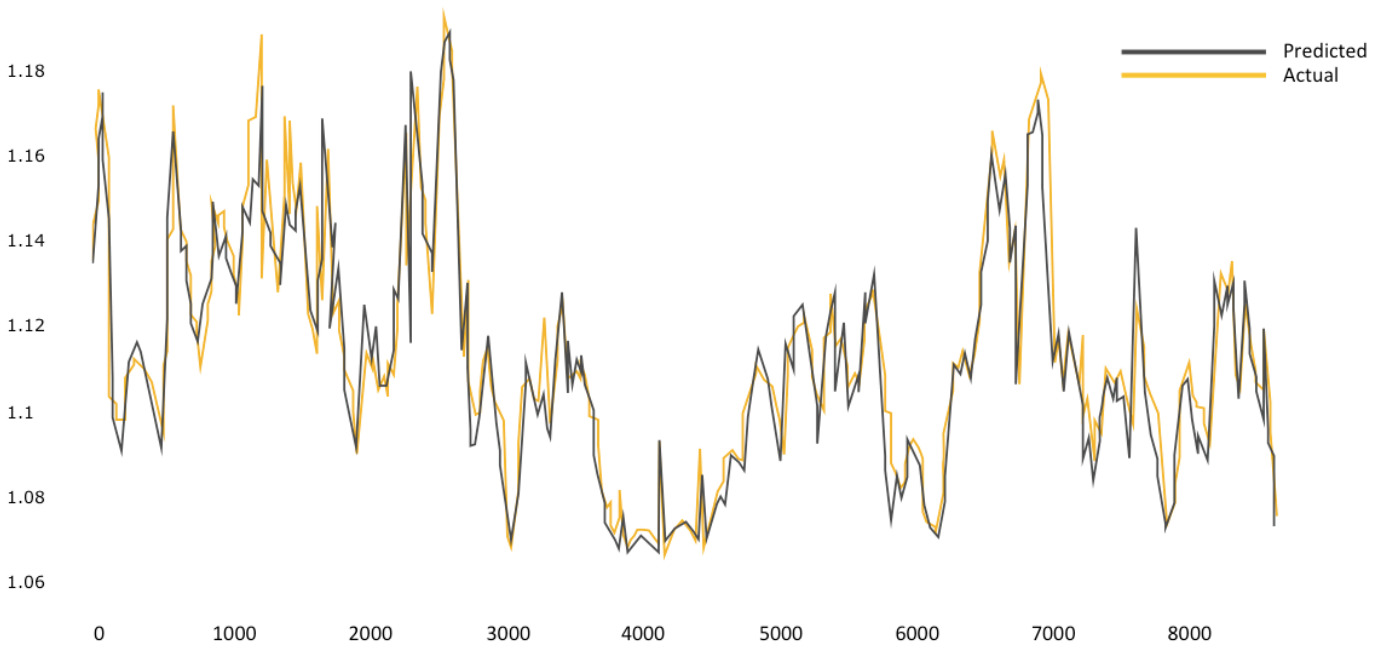
Les réseaux de neurones furent largement utilisés et au cœur de nombreuses recherches dans les années 80, puis leur popularité a diminué à la fin des années 90. Leur récente résurgence est due à leurs performances surpassant l'état de l'art pour de nombreuses applications.

Par exemple, le *Deep Learning* est le dernier outil utilisé par Google pour réduire la consommation énergétique de ses data centers. Pour ce faire, Google a enregistré les valeurs de 19 différentes variables – comme la vitesse des ventilateurs des data centers, la température extérieure ou encore l'humidité – toutes les 5 minutes pendant 5 ans. Ces données ont alors servi d'ensemble d'apprentissage à l'algorithme de *Deep Learning* dont le but est de prédire un score appelé PUE (*Power Usage Effectiness*) et qui renseigne sur la consommation énergétique globale du data center.



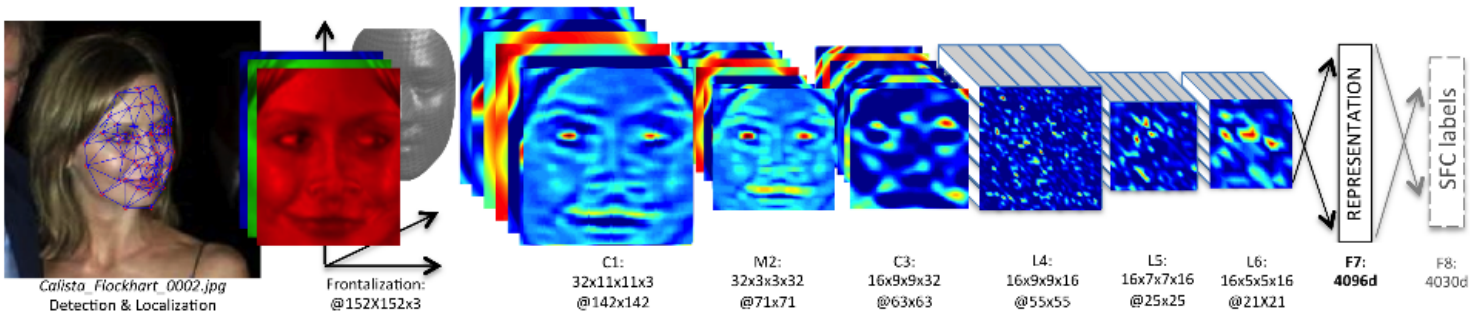
Le graphe ci-dessus est une représentation d'un réseau de neurones profonds (*deep*) comme ceux utilisés par Google. Chaque disque représente un neurone, et on observe les différentes couches du réseau avec notamment les couches intermédiaires dites cachées (*hidden*).

On peut alors apprécier la précision des prédictions grâce aux courbes suivantes, avec en jaune la courbe d'évolution du PUE réel et en noir la prédiction.



Ainsi, ces algorithmes parviennent à modéliser des abstractions de haut niveau à l'aide d'architectures profondes composées de multiples couches d'abstractions. On dit de cette méthode qu'elle apprend des représentations des données d'entrée, pour tenter de prédire au mieux la variable cible.

Pour illustrer cela, prenons l'exemple du traitement d'une image où le but est de répondre à la question « *Y a-t-il un visage humain dans cette image ? Si oui, de qui s'agit-il ?* ».



Cette tâche qui s'avère être simple pour un cerveau humain est en revanche très compliqué pour un ordinateur actuel. On représente usuellement une image par une matrice à plusieurs dimensions (YUV) où chacun de ses éléments correspond à un pixel de l'image. Mais peut-être n'est ce pas du tout une bonne idée de représentation de l'image si l'on cherche à répondre à la question précédente !

L'algorithme de *Deep Learning* va alors, à partir de l'usuelle représentation de l'image en pixel évoquée, apprendre de lui même une meilleure représentation pour la tâche désirée.

Sans entrer cette fois dans les détails mais pour insister sur la puissance et l'attrait actuel pour le *Deep Learning*, voici quelques grands projets récents autour de ce champ.

DeepFace : Il s'agit là du projet de Facebook de réduire le fossé entre les performances de l'homme et de l'ordinateur en terme de reconnaissance de visage. Les articles scientifiques parus récemment ont été de véritables percées dans le domaine.

Google Brain : Il s'agit là du projet d'expérimentation à grande échelle de Google où des *clusters* composés de 16000 CPU sont utilisés pour faire du *Deep Learning.* L'état de l'art dans la classification d'images standard a alors pu être améliorée de près de 70%.

Skype : Microsoft va proposer avant la fin de l'année un outil de traduction linguistique en temps réel à travers Skype, basé sur du *Deep Learning*.

Baidu : Le géant de l'internet chinois est en train de créer un immense centre de recherche autour du *Deep Learning*, dans la Silicon Valley, mené par Andrew Ng qui est un professeur de Stanford réputé pour ses travaux de recherche en *Machine Learning* et qui était auparavant à la tête du projet Google Brain.

Ainsi, on dispose d'algorithmes capables d'apprendre des données qu'on leur fournit. Et pour certaines tâches qui paraissaient inaccessibles pour un ordinateur il y a encore quelques années, certains algorithmes surpassent la précision humaine, comme pour la reconnaissance de caractères manuscrits.

Revenons à présent au commencement de mon stage.

Après avoir fait davantage connaissance avec les collègues avec qui j'allais passer mes six prochains mois de travail, il m'a été demandé de m'imprégner du travail de chaque équipe, de passer en revue ce qui avait été fait auparavant chez chacun des employés, et de proposer des idées de projets.

Cette phase a été extrêmement intéressante et enrichissante, car j'ai beaucoup appris dès le début en discutant avec toutes les personnes composant cette entité d'experts. Mais elle a surtout été surprenante dans la liberté et la confiance que l'on m'a accordées dès mon arrivée quant au choix du projet sur lequel j'allais travailler. Le fait que le projet me plaise et vienne de moi a toujours été important chez mes collègues qui se sont dès lors toujours assurés que j'étais épanoui quant à mon sujet de travail. Mon stage n'aurait alors pas pu mieux commencer, et cette agréable surprise n'aurait selon moi pas pu être possible dans une entreprise européenne.

Le management et la pression sont gérés d'une façon totalement différente de ce qui est imposé dans des entreprises en Europe ; cette dernière étant davantage axée sur les attentes, ce qui crée une espèce d'émulation générale permanente. Je trouve personnellement qu'il est beaucoup plus agréable et productif de travailler de cette façon, mais j'aurai l'occasion de revenir sur mon ressenti personnel dans la dernière partie de ce rapport.

Après de nombreuses discussions et l'étude de nombreux documents fournis, j'ai donc réfléchi à plusieurs idées de projets et préparé ma première présentation orale, avec les trois propositions de projet que j'ai personnellement choisi et dont voici une brève présentation :

1) Projets CDR :

Orange dispose d'une base de donnée appelée CDR (*Call Detail Record*) qui, comme son nom le suggère, archive l'ensemble des informations existantes lors des communications de ses abonnés. Par exemple lorsque l'appel d'un client est émis, on trouvera dans cette base de donnée notamment le numéro et donc le nom de l'abonné (et d'autres informations personnelles dont dispose éventuellement l'opérateur) ; mais aussi, et dans la limite du possible, les informations du destinataire de l'appel (suivant s'il est client chez le même opérateur ou non). Puis la date et l'heure de l'appel, sa durée, mais aussi la géolocalisation des protagonistes ou encore le type des appareils téléphoniques utilisés.

Cette mine d'or informative très riche laisse imaginer les potentielles et innombrables applications possibles pour quiconque y aurait librement accès. Mais ces données soulèvent éminemment la question brûlante qui ne cesse d'être dans l'actualité depuis quelques années et qui est évidemment celle de la protection de la vie privée de tout citoyen. Et d'autant plus avec tous les récents articles de presse impliquant Orange et la NSA ou la DGSE. Aussi cette base de donnée est très sensible et extrêmement bien protégée. Je pense d'ailleurs que les employés d'Orange disposant de l'accès à cette base de donnée sont très peu nombreux et qu'il est impossible pour une personne seule d'y accéder, peu importe son rang dans la société.

Xavier Quintuna, le Data Scientist à la tête du Big Data chez Orange Silicon Valley et qui oriente largement le groupe Orange dans ce domaine, est la personne avec qui j'ai le plus travaillé lors de mon stage. Il a alors bataillé pendant cinq ans pour obtenir une version totalement anonyme du CDR pour pouvoir faire des tests et lancer des projets autour de ces données. Il a finalement obtenu l'autorisation et a reçu une semaine d'enregistrements anonymes des communications des abonnés français d'Orange, ce qui représente tout de même 1TB en terme de stockage, avec 1 million d'enregistrements. Il a alors lancé depuis différents projets, seul ou avec des entreprises partenaires, qu'il m'a présentés au début du stage.

Cependant, Xavier n'avait jamais réellement tenté des projets utilisant du *Machine Learning*, ce qui m'a un peu surpris. C'est pourquoi j'ai pensé à un projet utilisant cette base de donnée. J'espérais en plus pouvoir croiser ces données avec celles résultant des navigations web chez les abonnés disposant de Smartphone, pour avoir un maximum d'information. Car l'opérateur dispose en effet de toutes les informations concernant les pages web visitées ou les mots clés recherchés, dès lors qu'un individu utilise son Smartphone.

A noter d'ailleurs que les opérateurs téléphoniques disposent ainsi d'une quantité et d'une qualité d'informations tout à fait significatives, à mon sens plus puissantes encore (ou dangereuses ?) que ce dont disposent Google ou Facebook.

J'ai alors proposé d'essayer de récupérer une base de données des clients se désabonnant (*churn*) pour tenter de le prédire avant que cela ne se produise, en observant l'immense quantité de données à notre disposition. Les problèmes qui se sont alors posés ont été ceux de récupérer cette base de données des désabonnés d'une part, et d'autre part nous ne disposions que d'une semaine de CDR, ce qui fait très court pour apprendre un comportement aussi complexe que celui de prendre la décision de changer d'opérateur.

J'étais en tout les cas résolu à vouloir tenter un projet avec le CDR, ne serait-ce parce que je n'étais pas sûr d'avoir de nouveau un jour l'opportunité de jouer avec une telle base de données. J'ai par exemple proposé de tenter d'identifier des *buying patterns*, soit des comportements dans les choix des clients pour créer des *clusters,* ou groupes, suivant les profils décelés. J'avais également proposé des applications pour détecter/prédire les opérations frauduleuses.

Xavier a cependant été davantage séduit par la dernière idée que je présenterai ci-après (le projet Davinci) et qui m'a incité à mettre le CDR de côté, peut-être un peu las de travailler sur ces données et désirant sans doute démarrer quelque chose de tout à fait nouveau.

2) Projets IOT :

Une des tendances technologiques majeures du moment est l'*Internet Of Things* (IOT), ou en français « l'internet des objets », « les objets connectés » ou encore parfois appelé « le Web 3.0 ». Orange Silicon Valley a d'ailleurs de nombreux projets autour de la question, avec des études de ce qui existe déjà, des tests, des développements de prototypes, et comme d'habitude des partenariats ou échanges avec les start-ups du domaine. Beaucoup d'entre elles traitent des *wearables* qui sont ces nouveaux petits objets portatifs, comme des montres, bourrés de différents capteurs destinés à nous suivre partout et qui fleurissent partout dans la *Silicon Valley*.

J'ai alors naturellement réfléchi à un projet sur ce sujet. Après avoir discuté longuement avec les personnes connaissant la question chez OSV, le problème était alors toujours celui des données à utiliser, peu importe l'idée sous-jacente. En effet le domaine étant assez jeune, il n'y a pas, ou très peu, de bases de données disponibles en relation avec le sujet. L'idée est alors de produire soi-même les données. J'ai ainsi proposé d'équiper l'*open-space* entier de capteurs lumineux et de détecteurs de mouvements afin de tenter d'apprendre une fonction optimisant l'intensité lumineuse individuelle des lampes d'éclairage.

Construire sa propre base d'apprentissage de la sorte est une tâche longue et qui coûte cher. De plus, l'engouement de la part de mes collègues comme de moi-même quant à l'idée suivante l'a emporté.

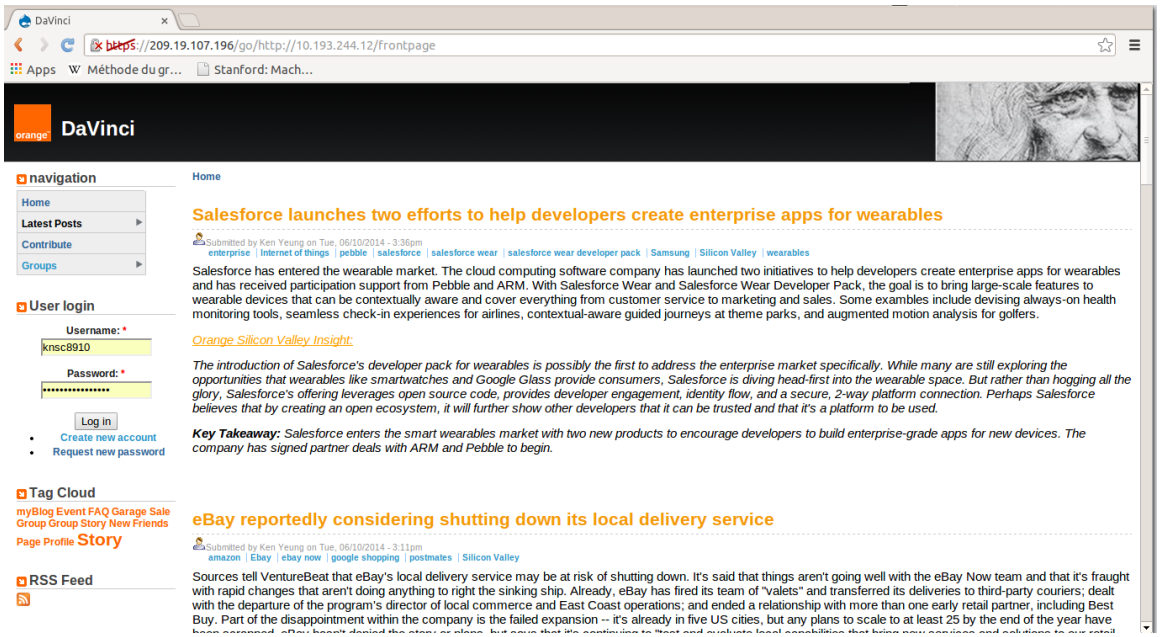
3) Projet Davinci :

C'est donc cet ultime projet qui a été retenu après cette première phase dans mon stage. L'idée de ce projet se base sur l'activité de veille stratégique d'Orange Silicon Valley.

Tous les jours depuis plus de 5 ans, tous les employés d'OSV, chacun expert dans un domaine technologique particulier, doivent parcourir les sources d'informations à notre disposition (essentiellement sur le web) et s'informer des nouveautés technologiques ayant lieu partout dans le monde.

L'orsqu'un événement est intéressant et en rapport avec l'expertise d'un employé quelconque, celui-ci se doit de résumer la nouvelle sur une plateforme interne appelée Davinci, et d'y ajouter un *insight* personnel. Le but est ensuite de trier les informations les plus pertinentes pour Orange et de créer chaque semaine une gazette envoyée vers la France, mais je reviendrai là dessus au moment d'aborder les missions secondaires du stage.

Voici à quoi ressemble l'interface en question. Il s'agit donc d'un outil collaboratif pour répertorier les *news* afin de pouvoir mieux gérer cette quantité d'informations et de pouvoir interagir sur les différents sujets abordés. On y distingue bien le dernier article posté, l'*insight* personnel qui le suit ; puis le début de l'article précédent.



Cette base de données, bien qu'assez petite en taille puisqu'elle ne pèse que 26Mo avec un total d'environ 16000 articles, est donc très informative et a vu émerger une quantité de nouvelles tendances relativement importantes depuis 5 ans, comme le *Big Data* ou le *Cloud Computing*, et c'est justement le point de départ du projet.

L'idée de base est alors de construire un système capable de prédire l'évolution des tendances, basé pour commencer sur la base de données Davinci. L'hypothèse raisonnable qui est alors faite est que les tendances technologiques sont bien reflétées dans cette base de donnée. Une seconde hypothèse est faite, comme nous allons le voir dans la suite, selon laquelle la quantité d'articles postés sur Davinci en relation avec un sujet reflète son importance en terme de tendance.

L'idée finale serait d'enrichir la base d'apprentissage avec des jeux de données économiques par exemple, ou bien d'écouter ce qui se passe sur un grand nombre de sites internet.

Nous allons voir dans la suite, pas à pas, ma façon de procéder sur ce projet.

La première tâche pour moi a alors été celle de créer des séries temporelles représentant l'évolution des tendances ; et ce pour chaque concept, technologie, start-up ou mot important. Le caractère quelque peu incertain et nouveau de la base de données Davinci m'a incité à construire et tester mes modèles en parallèle sur des times series reconnues par l'ensemble de la communauté scientifique comme étant une bonne base d'apprentissage pour tester des modèles. J'ai choisi de prendre les 111 séries de la compétion NN3 (*Neural Network competition*) disponibles gratuitement.

Modélisation

Prédire le futur est l'une des tâches les plus stimulantes pour les sciences appliquées. Construire des prédicteurs efficaces à partir de données passées requiert une certaine méthode statistique et computationnelle afin d'inférer des dépendances entre le passé et le futur à court terme, mais il faut aussi être capable de choisir la stratégie appropriée face à un futur à plus long terme et des horizons lointains.

Avant même de parler de tendance, il est utile de se fixer une définition pour ce terme. Une tendance concernant une série temporelle peut être caractérisée par une croissance ou une décroissance monotone, une tendance saisonnière (où on peut exiber un comportement cyclique, ou bien les deux en même temps. Mais certaines tendances ne sont pas facilement décelables par les modèles statistiques.

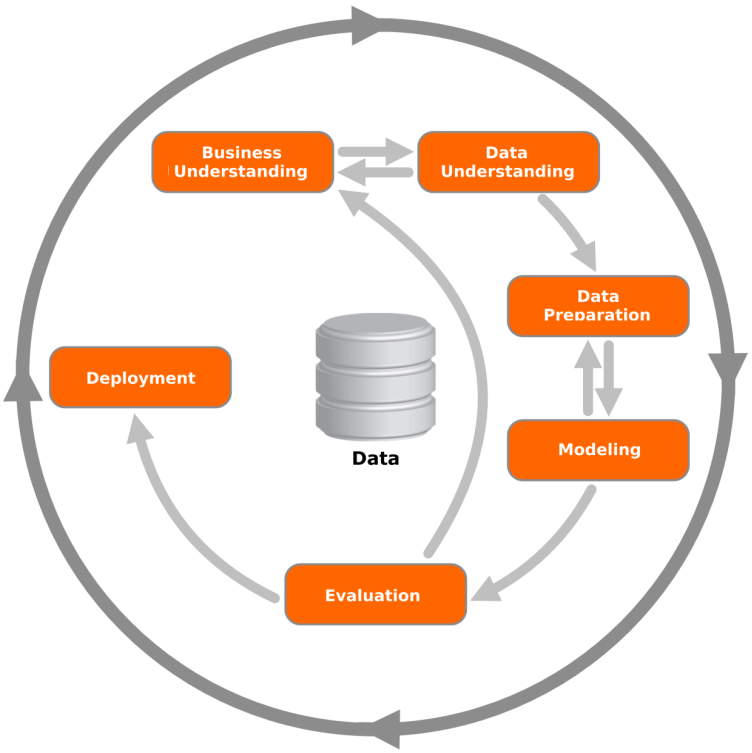
Lors des 20 dernières années, les modèles de *Machine Learning* ont attirés l'attention et sont même devenus plus attrayants que les modèles statistiques classiques utilisés jusqu'alors pour les tâches de prévisions par la communauté mathématique. Ils constituent des exemples de modèles non-paramétriques et non-linéaires qui se basent uniquement sur les données historiques pour apprendre les dépendances stochastiques existant entre le passé et le futur.

Des projets de recherche comme GIDA ou TREMA ont montré la forte demande de recherche et de développement de méthodes nouvelles et efficaces pour le *Trend Mining,* incluant des analyses d'informations textuelles.

Une fois le projet lancé, ma première tâche a alors été celle de lire de nombreux articles scientifiques, dont certains se trouvent dans la bibliographie de ce rapport, pour d'une part tenter de me mettre à niveau de l'état de l'art concernant la prédiction pour les séries temporelles ; et d'autre part pour être capable de décider de quelle façon préparer les données et de décider quel type de modèle je souhaitais tester.

On pourra trouver en annexe une étude plus théorique que j'avais rédigé en anglais pour certains collègues et qui explicite un peu plus les choix pris.

Mais passons tout d'abord rapidement sur la méthodologie générale du *Machine Learning*, et qui est celle que j'ai suivie.



La première des choses est de bien comprendre la tâche que l'on cherche à accomplir, et pour cela de bien comprendre les données sur lesquelles on se base pour y parvenir. Suite à cela, il faut préparer les données de la meilleure façon possible, afin qu'elles soient le plus informatives pour notre objectif prédictif. Vient ensuite la partie *Modeling* qui est selon moi la plus longue : il s'agit là de choisir quel modèle on souhaite essayer et surtout avec quels paramétres. On évalue alors le modèle entraîné sur un échantillon de test pour appréhender les performances du modèle. Le déployement vient alors seulement lorsqu'on estime ne pas pouvoir faire mieux pour chacune des différentes étapes.

Construction des séries temporelles

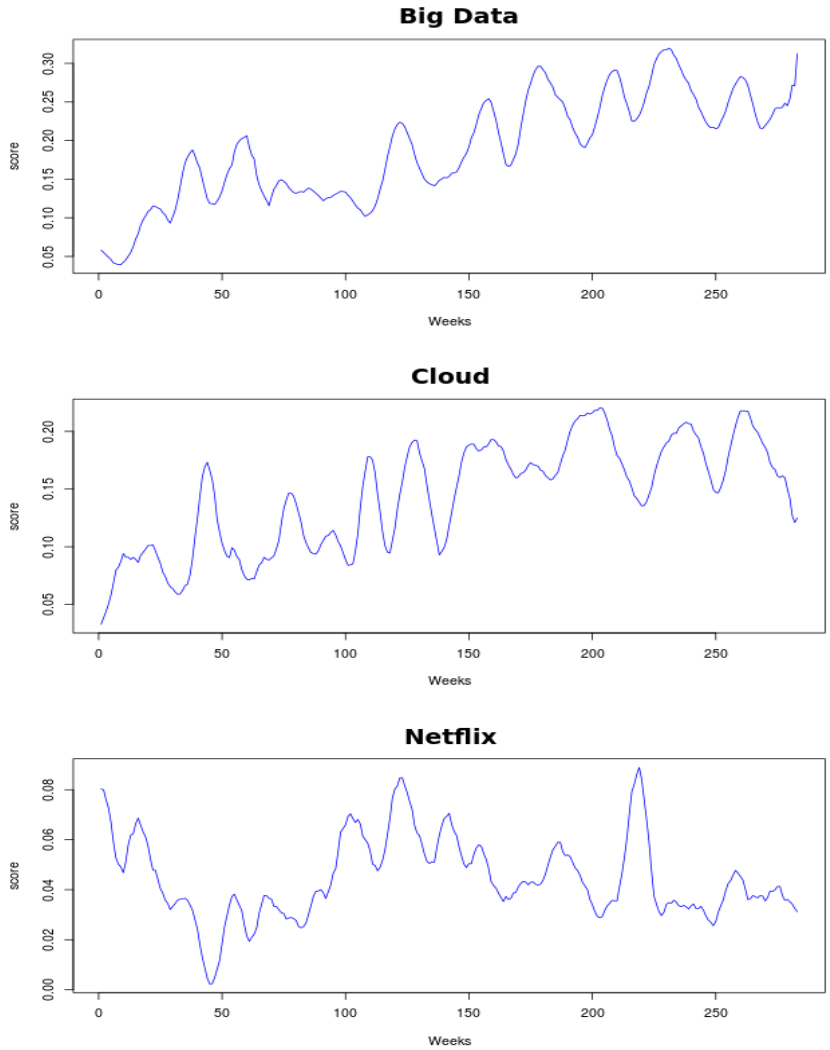
L'idée est donc ici de construire une série temporelle par tendance. La première étape pour cela a été celle d'extraire les mots clés de chaque article présent dans Davinci afin de connaître le sujet de celui-ci. J'ai pour cela utilisé l'algorithme TF-IDF qui est explicité en Annexe. J'ai développé la plupart de mes algorithmes sous le langage R.

Voici alors un nuage de mots (*worldcloud*) des occurrences les plus présentes.

On observe donc qu'il s'agit majoritairement de technologies et de noms de compagnies.

Une fois les mots clés extraits pour chaque article, reste à construire les séries temporelles en se basant sur leur nombre d'occurrence par semaine. Ainsi la période temporelle choisie pour les séries a été dans un premier temps la semaine.

Voici par exemple l'allure des trois séries temporelles *Big Data, Cloud* et *Netflix*.



Vient ensuite la partie d'entraînement des algorithmes de *Machine Learning*, et la sélection de modèle visant à estimer la performance des différents algorithmes afin de choisir le meilleur possible.

Cette partie est de loin la partie la plus longue en terme de temps de calcul. Les calculs prennaient régulièrement plusieurs jours avant de pouvoir enfin comparer les modèles obtenus.

C'est seulement une fois qu'un modèle performe bien sur un échantillon test qu'on peut l'utiliser pour prédire l'évolution future des différentes tendances.

Autre application : Prédiction de la valuation des *Unicorns*

En plus de chercher à prédire l'évolution de ces tendances, mon projet a pris une seconde dimension lorsque l'on m'a proposé d'essayer de prédire la valuation d'un certain type de start-ups appelées *unicorns* en prennant comme données d'entrées les mêmes séries temporelles précédentes, mais uniquement celles concernant les *unicorns*.

Ces *unicorns* correspondent à un nouveau genre de start-ups regroupant celles qui connaissent une croissante fulgurante. Citons l'exemple de WhatsApp qui s'est fait acheté 19 milliards de dollars US par Facebook en Février 2014, alors qu'elle était jeune d'à peine 5 ans.

On dira qu'une start-up est une *unicorn* si celle-ci reçoit une valuation de plus d'un milliard de dollars US par des investisseurs privés, les marchés publics ou pour l'acquisition par une entreprise. On trouvera en annexe la liste des principales unicorns avec lesquelles j'ai travaillé, classées par domaine d'activité.

J'ai alors isolé les séries temporelles concernant ces *unicorns,* et la variable à prédire était cette fois la valuation de celles-ci au jour du 10 Avril 2014.

La valuation d'une start-up est une variable extrêmement complexe, qui dépend d'énormément de facteurs. Il m'a paru dès le départ très compliqué d'espérer avoir des prédictions intéressantes, d'autant plus que les séries temporelles dont je disposais semblaient assez peu informatives, comme on peut le voir sur les courbes suivantes. Cela est dû aux caractéristiques même des *unicorns* : leur croissance rapide et leur jeunesse. Mais comme on le verra dans la partie tournée vers les résultats, cela vallait le coup d'essayer.

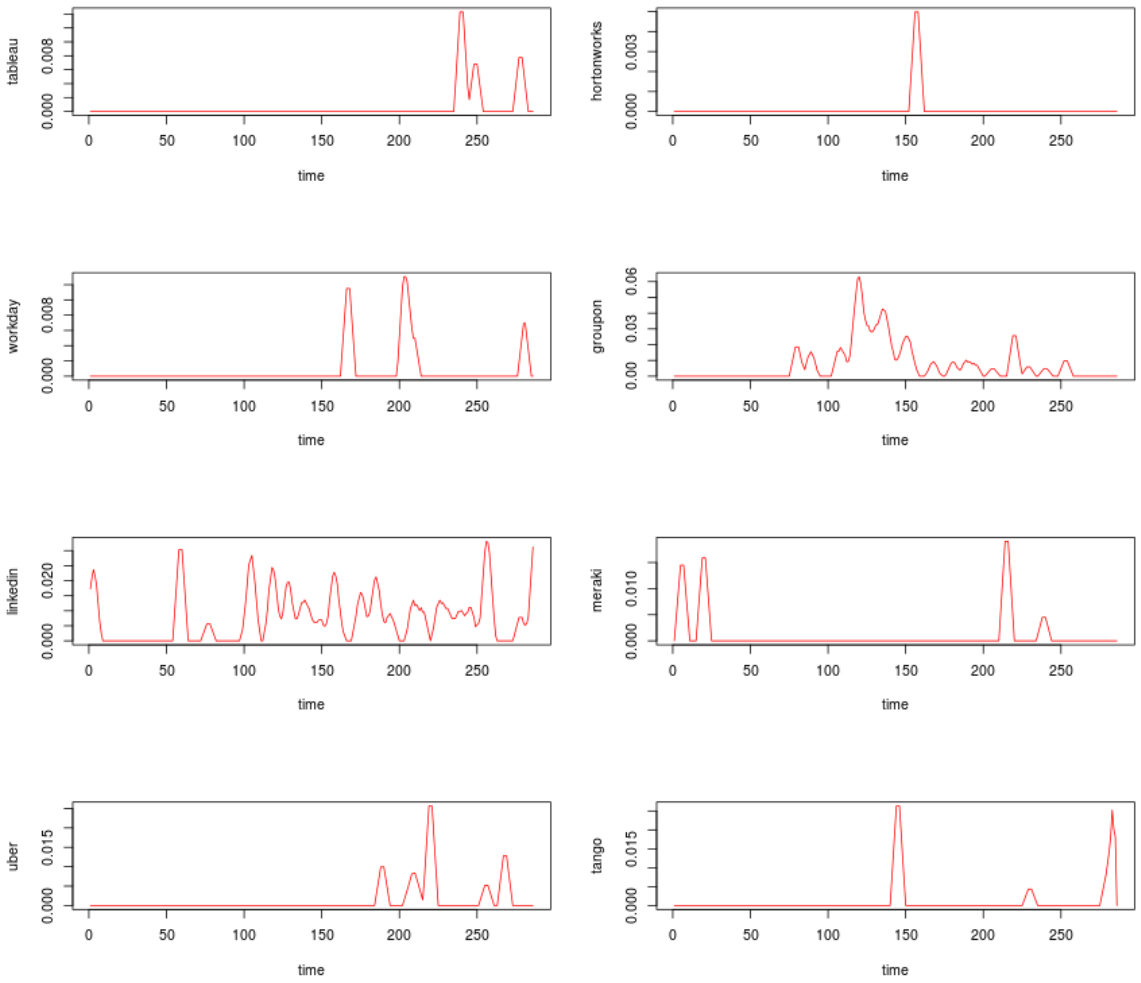


tableau
0.000
0.008
0
50
100
150
200
250
time
hortonworks
0.000
0.003
0
50
100
150
200
250
time
workday
0.000
0.008
0
50
100
150
200
250
time
groupon
0.00
0.03
0.06
0
50
100
150
200
250
time
linkedin
0.000
0.020
0
50
100
150
200
250
time
meraki
0.000
0.010
0
50
100
150
200
250
time
uber
0.000
0.015
0
50
100
150
200
250
time
tango
0.000
0.015
0
50
100
150
200
250
time

2.2 Mes autres activités

Comme déjà expliqué longuement, rendre compte des tendances de la Silicon Valley est une tâche quotidienne et qui concerne tous les employés d'OSV, stagiaires compris. Le procédé est en fait un tout petit peu plus élaboré que la simple existence de la plateforme Davinci.

Les « News room »

Tous les jours, mon groupe « *Middleware and Platform* » se réunissait afin de discuter des *news* de la journée. En effet, nous devions donc parcourir des articles de presse (ou plutôt parcourir le web) afin de dénicher des *news* ou des évènements relatifs à notre domaine de compétence.

Ensuite, nous nous réunissions tous les jours à 15h, afin de débattre sur toutes les *news* dénichées par le groupe.

Ainsi, nous pouvions donner notre avis et répondre à des questions du type:

- Quelles conséquences pour le marché ?
- Quelles opportunités pour Orange ?
- Quelles sont les initiatives que nous envisageons en fonction de cette information ?
- Un partenariat est-il possible ?
- Achat de services ou de produits.

Ensuite, une fois que tous les membres du groupe avaient présenté leurs *news* nous devions sélectionner la *news* du jour. Afin que la personne la plus à même d'analyser cette *news* rédige un article dans la « gazette ».

Ce principe de *news room* a été très intéressant, car il nous obligeait à rester très au fait de l'actualité technologique tout en soudant l'équipe avec des réunions quotidiennes. J'ai ainsi énormément appris par moi-même en lisant les *news* quotidiennement mais également grâce aux connaissances partagées durant ces moments d'échanges.

Cependant, l'heure choisie pour cette réunion ne fut pas toujours idéale et a pu constituer une coupure perturbante dans l'après midi parfois dérangeante lorsque que l'on souhaite avancer sur son projet.

Après la réorganisation d'OSV, de nouveaux groupes ont été créés pour les *news rooms* qui fonctionnaient alors davantage par projet, comme tout le reste.

Les articles de la gazette

Lorsque le groupe avait décidé que telle *news* était la plus intéressante de la journée, une personne était chargée de rédiger un article succinct de la news. Cet article devait comprendre une description de la news et le plus important : un avis personnel (ou « *insight* »). En effet, nous devions prendre suffisamment de recul par rapport à la *news* et tenter de donner un avis éclairé sur celle-ci. Et ainsi, expliciter en quoi elle était importante pour Orange et quelles en seraient les conséquences. Le tout en moins de 300 mots.

La rédaction de cet article n'était pas toujours évidente au début, car il fallait rédiger un article de 300 mots correctement tournés en anglais. Mais c'était l'occasion unique de donner son avis sur des faits et également d'être lu par le top management dont le CEO: Stéphane Richard.

En effet, chaque semaine, une équipe est chargée de mettre en forme l'ensemble des articles de tous les groupes, et en se basant également sur Davinci, pour en faire un seul et unique document qui est ensuite envoyé aux hauts dirigeants. Ces articles permettent alors de se rendre compte assez rapidement des tendances actuelles aux Etats-Unis, et grâce aux commentaires de participer à l'élaboration de stratégies pour le groupe.

Ainsi, entre la lecture des *news* et la rédaction des articles dans Davinci, je passais près de deux heures par jour sur cet objectif. Par contre, ce qui était relativement frustrant, c'était de n'avoir aucun retour des réactions des hauts dirigeants quant aux différents avis émis. Il nous arrivait même de nous demander si ces gazettes étaient vraiment toujours lues, mais en tous les cas, ces heures étaient très instructives.

Rencontrer des entreprises

Lors des *news room*, nous ne nous contentions pas uniquement de rédiger des articles. Parfois lorsqu'une entreprise nous semblait innovante ou intéressante à suivre, nous prenions contact avec celle-ci afin de faire connaître Orange dans la Silicon Valley et d'établir parfois un partenariat commercial.

Avec le poids d'Orange dernière lui, son sens humain et son bagou, Xavier devient rapidement proche de toute les start-ups dans le domaine de la data. J'ai alors eu l'immense chance de pouvoir rencontrer de nombreuses start-ups parmi les plus importantes du moment dans le milieu, et d'avoir une relation privilégiée avec les CEO ou les employés de ces futurs géants.

En effet, Xavier m'a toujours présenté comme le « *data scientist in house* » d'OSV et m'a toujours beaucoup mis en avant. J'ai d'ailleurs moi même contacté quelques entreprises afin d'intégrer leurs solutions dans mon projet. J'ai par exemple directement travaillé avec l'entreprise OxData lorsque j'avais des questions sur l'utilisation de leur produit H2O.

J'ai également rencontré plusieurs fois la start-up Concurrent dont j'ai utilisé *Cascading*, leur couche d'abstraction logicielle pour Hadoop, technologie que j'ai appris à utiliser pour mon projet principal, ce sur quoi je reviendrai par la suite. Cette entreprise a des clients comme Airbnb ou Twitter, c'est dire à quel point ce ne sont pas des amateurs. Je pourrais également citer Interana, une start-up que j'ai rencontré plusieurs fois et qui propose un outil d'analyse pour de gros volumes de données ou *Big Data*, travaillant par exemple avec ebay. Ses co-fondateurs étaient parmis les tous premiers à travailler avec Mark Zuckerberg sur Facebook.

Il était alors toujours génial de pouvoir assister aux démonstrations de ces outils dernière génération et de pouvoir ensuite les tester.

A la toute fin de mon stage, j'ai aussi moi même fait se rencontrer Xavier et une jeune entreprise selon moi très prometteuse : Dataiku. Xavier les a d'ailleurs envoyé vers les bonnes personnes chez Orange France et désire travailler avec eux pour tester leur produit. Il s'agit d'une entreprise que j'ai découverte lors de mon dernier meetup. J'avais alors pu rencontrer le cofondateur français et discuter avec lui. L'idée de l'outil proposé par Dataiku est de permettre aux Data Scientists de traiter, nettoyer, croiser et créer des modèles prédictifs sur des grands volumes de données brutes, de façon automatisée et avec des interfaces visuelles pour les analystes et des APIs pour les développeurs.

La présentation du cofondateur m'avait beaucoup plus, il avait par exemple expliqué comment en utilisant leur plateforme, une équipe de Data Scientists a pu récemment gagner une compétition Kaggle. J'ai été alors content d'être inviter à leur rendre visite à Paris.

Les conferences et les meetups

Comme je l'ai déjà écris, j'ai parfois eu la sensation de faire un travail de journaliste. C'était particulièrement le cas lorsqu'il y avait des sommets ou des événements importants dans le monde de la *tech* et que nous devions couvrir.

Par couvrir, j'entends ici être sur place (peu importait d'ailleurs le prix des billets qui pouvaient parfois s'élever à plusieurs milliers de dollars US), capter le maximum d'informations et discuter avec le plus de personnes possible, et faire ensuite un rapport.

Il pouvait aussi très bien s'agir de petits événements. Du moment que l'on était motivé et que l'événement vallait le coup, on pouvait y aller presque à coup sûr. Je me suis par exemple rendu, avec deux collégues stagiaires, à l'université de Stanford pour assister à 80 *pitchs* de projets d'étudiants en fin d'étude à la *Design School*. Chaque équipe, composée de 2 ou 3 étudiants, devait développer une application mobile, préparer une affiche et un pitch d'une minute.

Nous avons pu discuter avec de nombreuses équipes. Les sujets qui sont ressortis le plus souvent tournaient autour des thémes suivants : « être plus productif en organisant son temps », « être connecté avec ses amis ou avec de nouvelles personnes » et « vivre plus sainement ».

Je suis retourné une seconde fois à Stanford, cette fois pour une visite guidée de l'université organisée par OSV pour ses stagiaires. Nous avons notamment visité le laboratoire d'intelligence artificielle, où travaille Andrew Ng. Nous avons pu tester de très nombreux prototypes de robots et d'appareils de réalité augmentée.

Recrutement Orange Fab

J'ai également eu la chance de participer à la nouvelle saison de recrutement d'Orange Fab, l'accélérateur de start-ups d'OSV. De nombreuses start-ups étaient candidates, et elles ont été réparties suivant les différentes équipes. Nous devions alors sélectionner lesquelles nous paraissaient les plus prometteuses après les avoir scrupuleusement étudiées. Parmi les start-ups sélectionnées, seules cinq auront l'opportunité d'être aidées par Orange Fab, après l'épreuve de la présentation orale devant un jury qualifié qui prend le temps de connaître les potentiels futurs vainqueurs qui cherchent à ne pas se tromper. J'ai eu la chance d'assister à une partie de ces *pitchs*.

Découvrir de nouveaux outils

L'objectif de mon stage tel qu'il était décrit dans ma feuille de route (*training plan*) envoyée une fois que j'avais été accepté était celui de decouvrir et d'apprendre à maîtriser différents outils de data science, notamment autour de Hadoop.

Haddop est devenu rapidement un outil standard pour paralléliser des traitements informatiques pour de grosses quantités de données sur un groupe d'ordinateurs. Un grand nombre de gros projets orbitent autour de Hadoop et utilisent cet outil. Certains ont pour but de manager les données, d'autres de visualiser et surveiller les processus en cours, d'autres encore de proposer des outils de stockage sophistiqués.

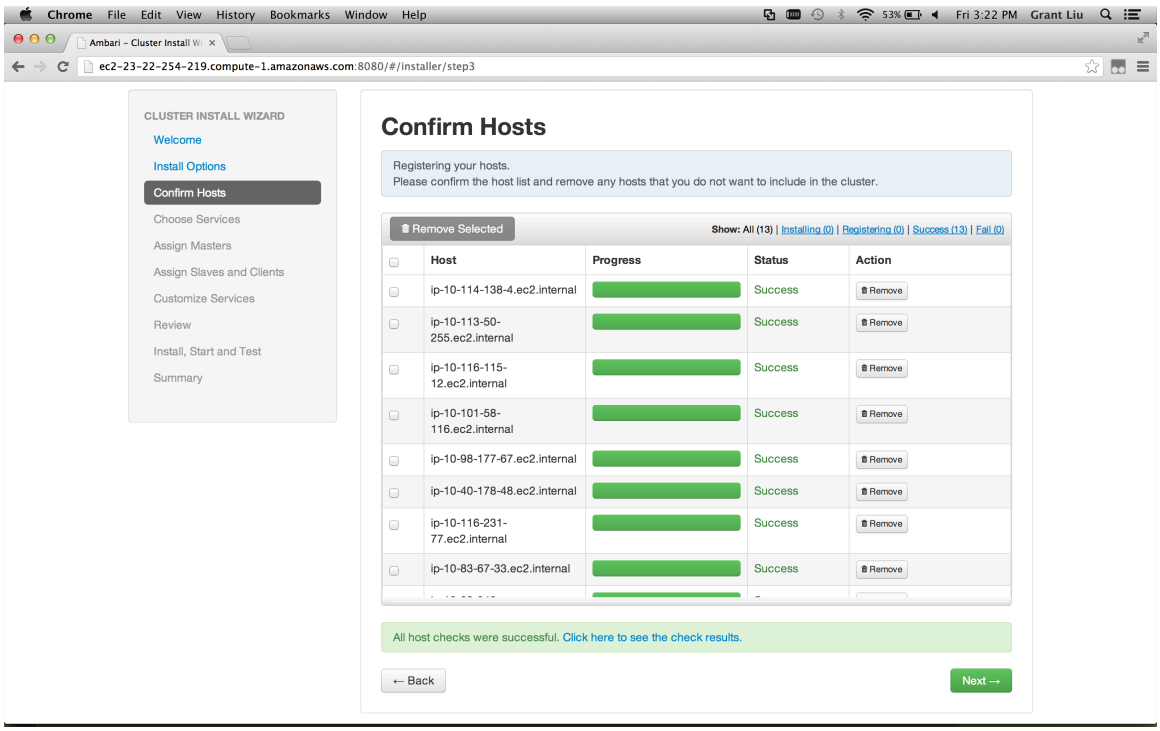
Un peu plus précisément, Hadoop est un *framework* développé en Java et qui a pour but de synchroniser des nœuds (un ordinateur est un nœud) esclaves qui exécutent une fonction codée d'une certaine manière sur les données qui sont stockées localement sur ce nœud. Les résultats des fonctions de chaque nœud sont ensuite agrégés puis traités par un nœud maître. La première opération porte le nom de « *map* » et la seconde de « *reduce* ». Le but est évidemment de pouvoir utiliser tous les ordinateurs en même temps pour une tâche donnée, ce qui permet de gagner beaucoup de temps et de traiter de grosses bases de données.

Hadoop offre alors une abstraction concernant la syschronisation et le déplacement des données traitées sur chaque nœud. Les programmeurs ont simplement besoin d'écrire leur code selon le paradigme « *map-reduce* » pour analyser les données, le travail est ensuite divisé et ordonné par Hadoop. Avec de grosses quantités de données et donc une grosse quantité de calcul, des erreurs peuvent apparaître frequement, mais Hadoop est justement conçu pour pouvoir faire face à des erreurs sur des machines individuelles, et c'est justement là toute la puissance de l'outil.

La communauté Hadoop a grossi très rapidement et de nombreuses plateformes payantes utilsant Hadoop ont vu le jour et permettent de manager des *clusters*, et permettant de grosses avancées dues au fait que tous ces projets sont *open source* et donc collaboratifs.

Je vais tenter de présenter succintement les quelques outils en relation avec Hadoop que j'ai pu découvrir et souvent utiliser.

Ambari : Mettre au point un cluster Hadoop peut impliquer de nombreuses tâches répétitives. Ambari offre une interface web utilisateur permettant de gérer les paramètres des principaux composants du cluster. Manager et surveiller les tâches effectuées par le cluster devient alors plus facile. On peut voir sur l'image ci-dessous à quoi ressemble l'interphace après avoir lancé un *cluster*.



HDFS signifie Hadoop Distributed File System et constitue une composante primordiale pour Hadoop. Il s'agit de l'outil dont la tâche est de diviser puis distribuer la base de données à traiter sur les multiples noeux du cluster, il se charge également de répliquer chaque sous ensemble de données créé pour palier à une éventuelle erreur de la part d'un nœud quelconque, chose évoquée précedemment. Ainsi une grosse base de données est séparée en plusieurs blocs qui seront distribués sur plusieurs nœuds.

Le système est donc conçu pour faire face aux erreurs et est très robuste. On comprend aussi que ce système a été designé pour des tâches lourdes en terme de temps de calcul, dans le sens où il devient obsolète si le temps nécessaire pour déplacer les données est plus long que le temps de calcul sur chaque nœud où les données sont stockées momentanément localement.

HBase : Il s'agit là du gestionnaire de base de données construit par dessus Hadoop, conçu donc pour de très grosses bases de données qui deviendront alors traitables par le *cluster*.

Hive : Mais injecter les données dans le cluster n'est que le début des réjouissances. Hive est un langage très similaire au SQL, conçu pour extraire de l'information des tables de HBase. Ce langage est également conçu au dessus de Hadoop, dans le sens où il abstrait totalement les tâches *map-reduce* qu'il exécute en réalité.

Sqoop : Il s'agit là d'un outil servant par exemple à charger des bases de données SQL traditionnelles sur une infrastructure Hadoop, et pour qu'elles passent sous le contrôle des outils Hive ou HBase.

Pig : Une fois les données stockées locallement sur chaque nœud du cluster, les choses sérieuses commencent. Pig est un langage de programation qui, de nouveau, abstrait les tâches *map-reduce* et laisse relativement de libertés ; du moment que les algorithmes peuvent être parallélisés sur le cluster. Des fonctions pré-existent, mais l'utilisateur peut très bien créer ses propres fonctions.

ZooKeeper est un outil qui permet de synchroniser, hiérarchiser et ordonner les nœuds entre eux suivant les différentes tâches que l'on désire effectuer sur le cluster et devient très utile pour des clusters comportant de nombreux nœuds.

Mahout est un projet designé pour proposer des algorithmes de *Machine Learning* en utilisant directement Hadoop. Malheuresement, les réseaux de neurones ne sont pas encore proposés, je n'ai donc pas utilisé cet outil.

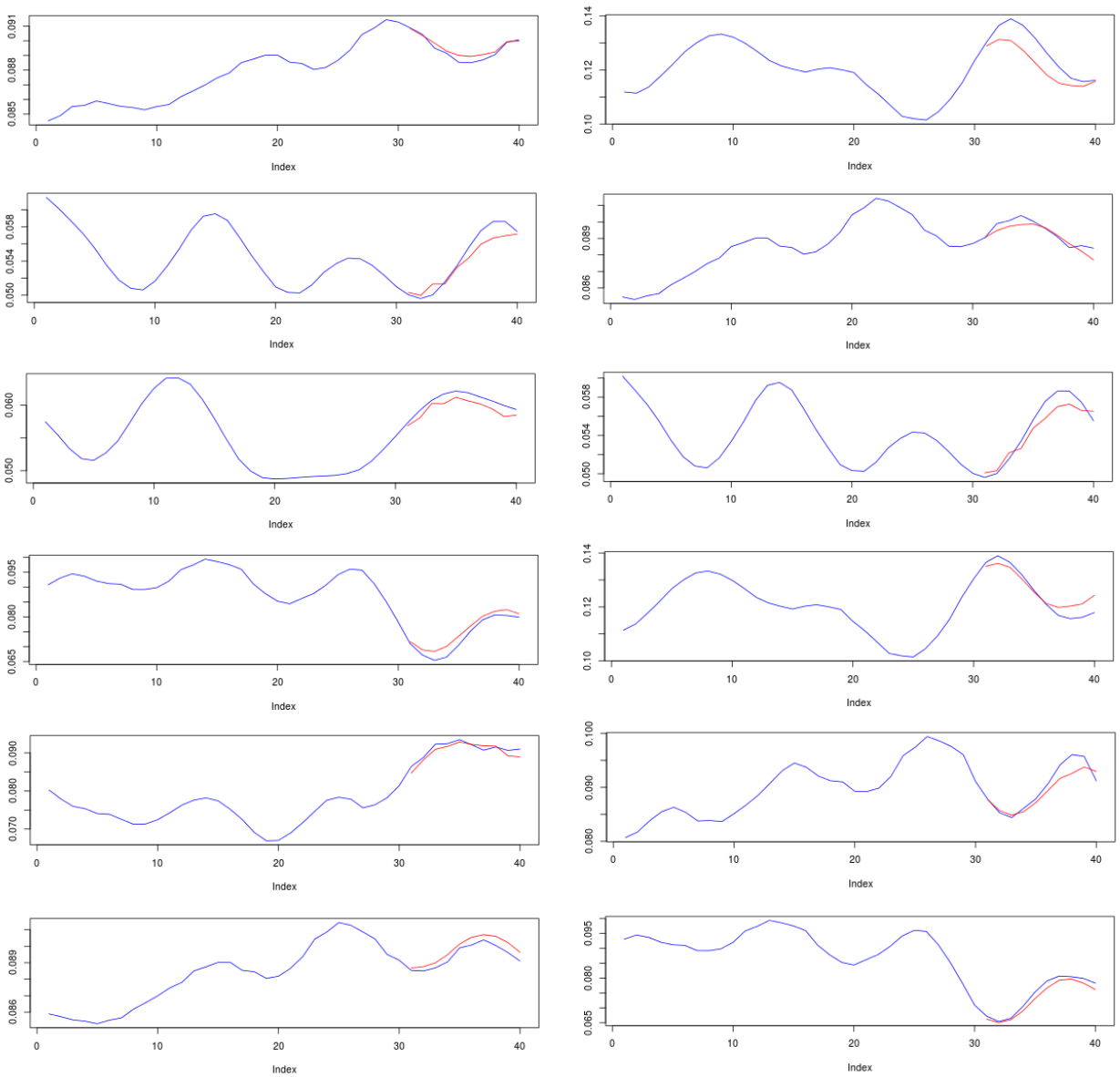
Oozie permet de lancer différentes tâches sur le cluster dans l'ordre désiré, sans avoir à attendre qu'une tâche se termine avant de pouvoir lancer la suivante. Il suffit de créer un graphe orienté acyclique avec les différentes tâches, et de lancer le tout.

Flume est un outil permettant de récupérer facilement des informations, par exemple sur internet, pour les charger ensuite dans HDFS.

Spark est sans doute le futur de Hadoop. Pour certains algorithmes, Hadoop peut s'avérer être lent car son fonctionnement repose sur le transfert et le stockage des données sur les disques des différents nœuds. Lorsque de nombreux appels sont faits sur les données, comme c'est le cas par exemple pour certains algorithmes de *Machine Learning,* les temps de chargement et d'écriture peuvent devenir longs comparés aux temps de calcul. Spark est alors la nouvelle génération fonctionnant sur la même idée que Hadoop mais avec les données chargées en mémoire.

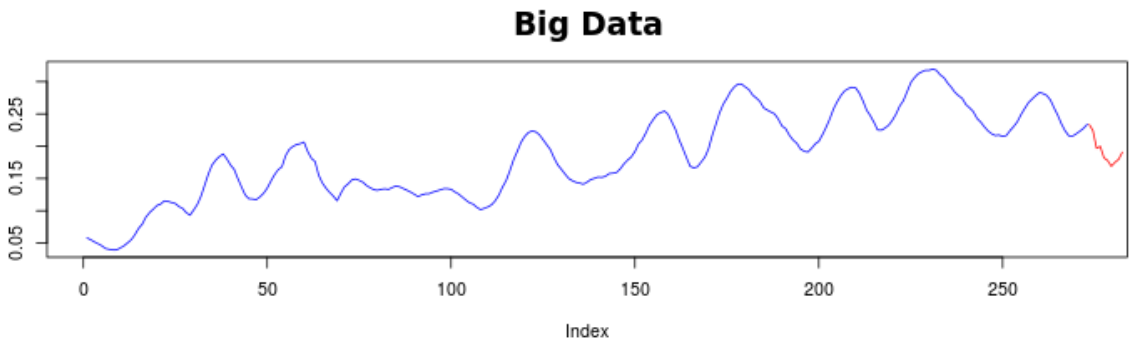
2.3 Les résultats obtenus

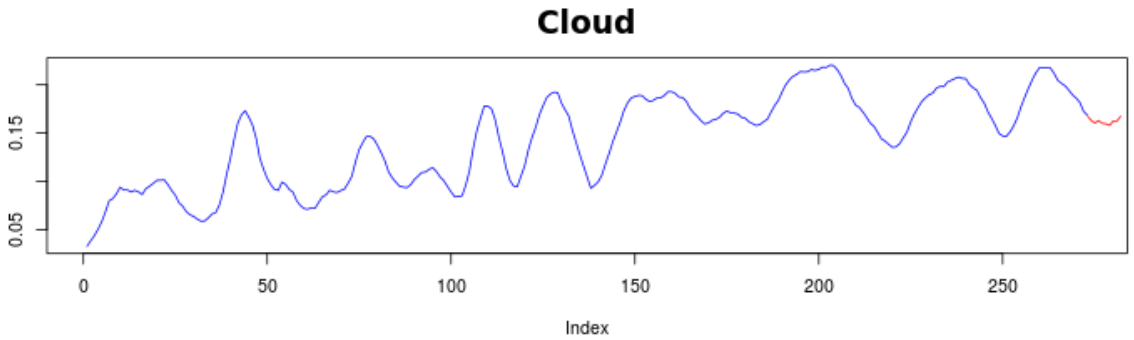
La phase de sélection de modèles visait donc à déterminer les hyper-paramètres les meilleurs possibles. Voici par exemple le résultat graphique des prédictions d'un modèle ayant été jugé bon par le processus de sélection :

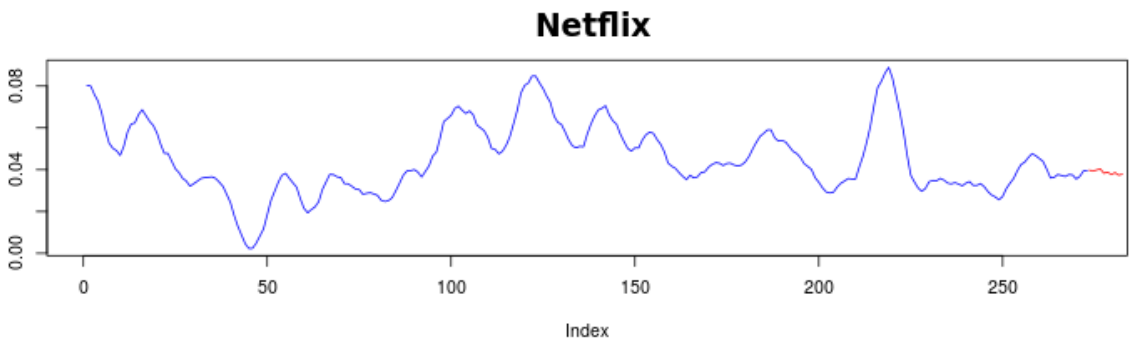


On observe alors les courbes bleues qui correspondent aux valeurs réelles des courbes de tendances échantillonées aléatoirement et d'une durée de 40 semaines ici. Les 30 premières valeurs sont alors données comme entrée à l'algorithme entraîné, les 10 dernières lui sont cachées et c'est ce qu'il doit prédire s'il performe bien. On observe alors les prédictions retournées en rouge, et on constate en effet que le modèle fait de bonnes prédictions.

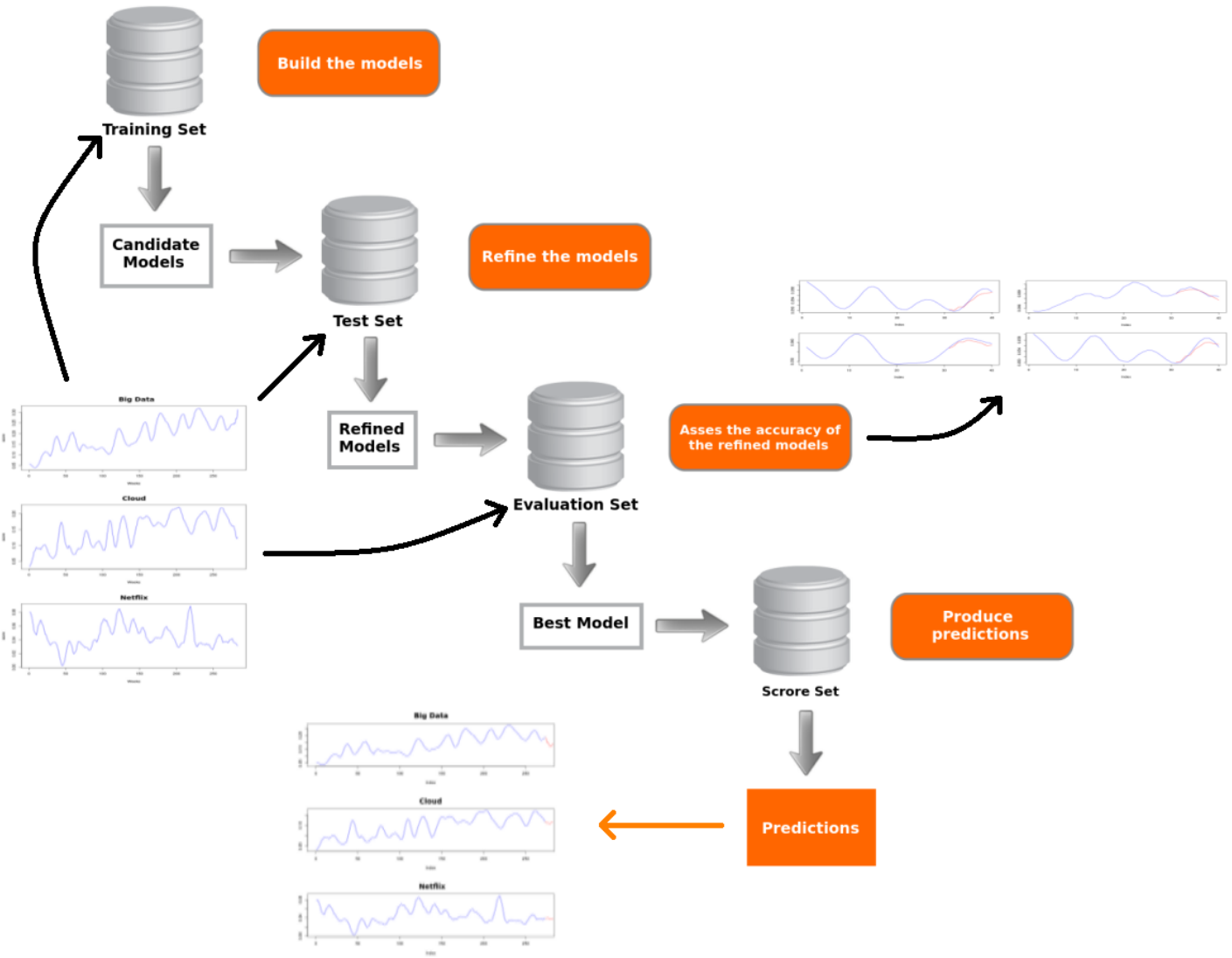
La phase de test étant passée et le modèle validé, on peut désormais utiliser celui-ci pour faire des prédictions futures cette fois ci, en utilisant par exemple ici en entrée pour l'algorithme les 30 dernières valeurs des séries temporelles des tendances. Dans ce cas, on ne prendrait donc pas en compte les semaines précédentes qui ne serviraient qu'à fournir l'échantillon d'apprentissage de l'algorithme. On peut alors dans ce cas obtenir ce type de prédictions :





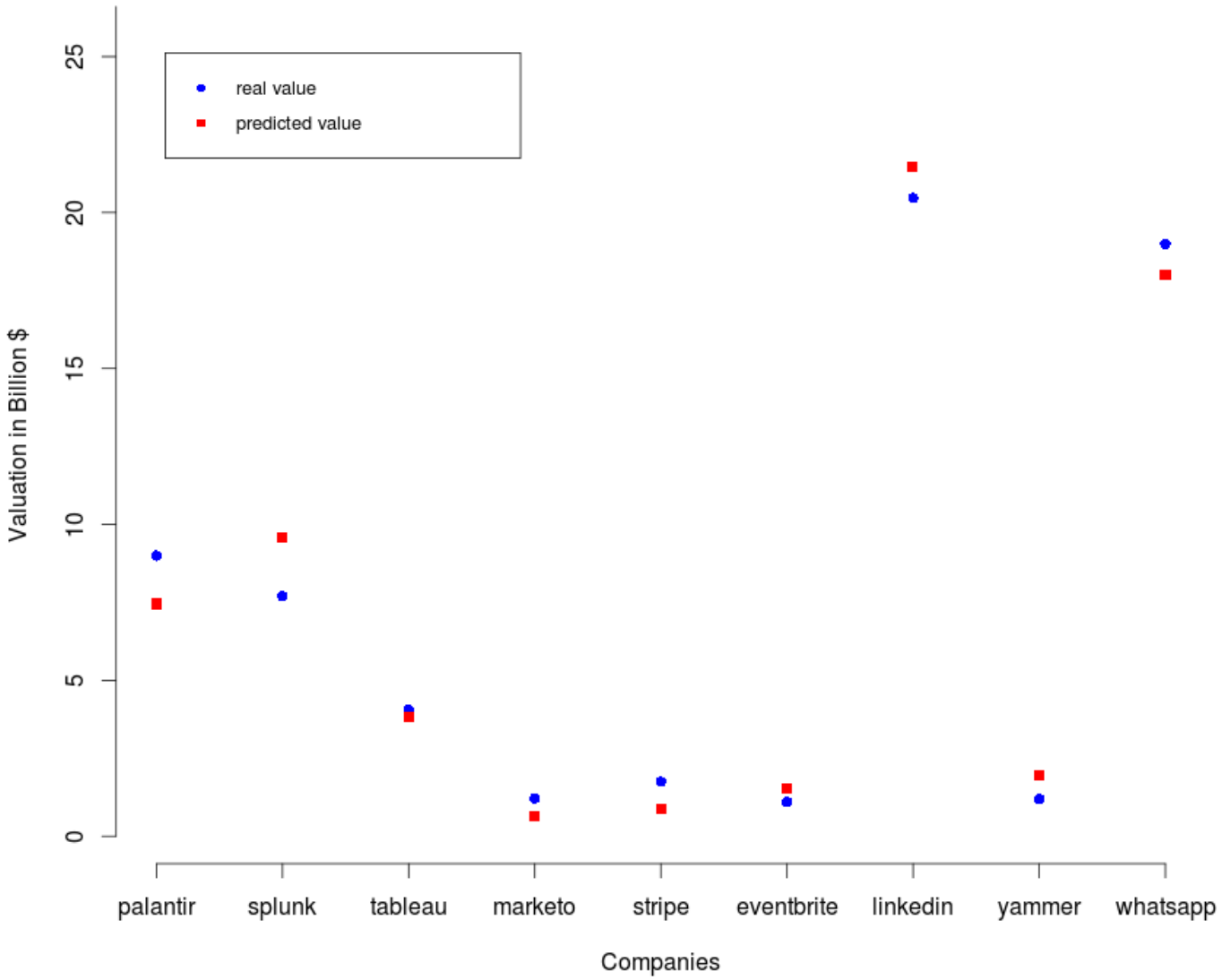


On peut résumer le processus d'entrainement et validation du modèle à l'aide du graphe suivant.

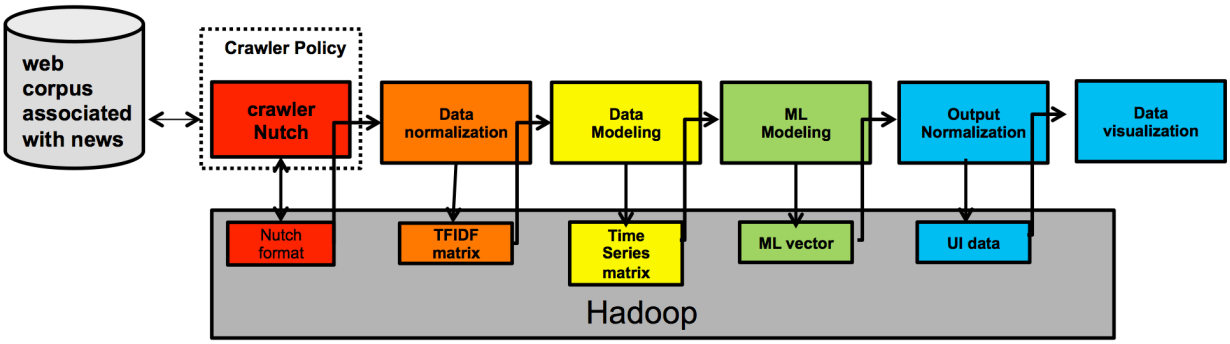


Pour ce qui est du projet *Unicorn*, je ne m'attendais pas à des résultats probants compte tenu de la complexité de la variable à prédire, à savoir la valuation des start-up ; et compte tenu également que les séries temporelles sur les *unicorns* semblaient contenir relativement peu d'informations (à savoir de longues périodes où les valeurs sont nulles).

J'ai été alors quelque peu surpris lorsque les prédictions se sont avérées ne pas être ridicules, comme on peut le voir sur le graphe suivant.

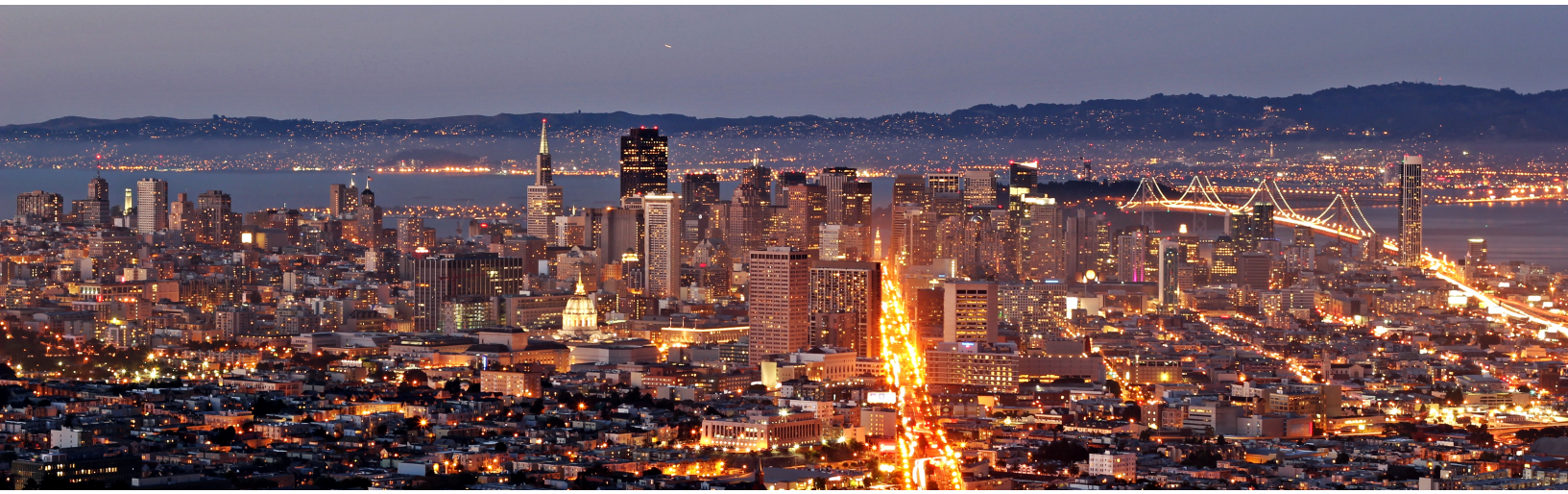


Avant mon départ, ma dernière tâche a été de bien expliquer ma façon de procéder à différents collègues dont Xavier qui souhaite poursuivre le projet en écoutant cette fois tout ce qui se dit sur différents sites d'informations concernant le monde *High Tech.*



J'ai fait de nombreuses présentations lors de mon stage, à des collégues notamment, mais aussi au Vice Président d'OSV, et surtout à Odile Roujol qui est la Directrice de la Stratégie Client et Data chez Orange. Celle-ci était venu au mois de Juillet avec le top management en ce qui concerne la data chez Orange (en tout 7 personnes) pour rencontrer des start-ups de la Silicon Valley et assister à quelques présentations d'OSV. J'ai alors eu la chance de leur présenter mon projet ainsi que les introduire au *Deep Learning*. Ma présentation leur a d'ailleurs beaucoup plu et nous sommes restés ensuite un long moment pour discuter.

Xavier a émi pour la première fois l'idée que je reste en *full time* après mon stage seulement 3 mois après mon arrivée, soit au milieu de mon stage, ce qui est assez gratifiant. Il m'en a dès lors reparlé peu à peu mais j'ai pris la décision de ne pas rester car j'avais d'autres projets pour l'an prochain. Je dois dire que j'ai énormément ressenti la demande de *Data Scientists* par les entreprises *tech* et le manque de personnel qualifié car dès lors que j'ai mis à jour mon profil Linkedin au début de mon stage, j'ai commencé à recevoir des offres de postes à raison d'une par semaine en moyenne, en général par des entreprises de la Silicon Valley dont certaines grosses entreprises comme Apple, mais aussi des offres en régions parisienne.



Partie 3 :
Difficultés rencontrées et apports du stage

3.1 Faits marquants et difficultés rencontrées

La toute première des difficultés qui auront été les miennes et qui me vient tout de suite à l'esprit est celle de la langue. En effet en arrivant pour la première fois aux Etats-Unis, j'avais un niveau d'anglais moyen, avec des difficultés à m'exprimer. Je n'avais alors jamais passé de longue période dans un pays anglophone. Bien que la communauté française soit très développée à San Francisco (j'ai d'ailleurs eu le plaisir de rencontrer de très nombreux étudiants en stage ou VIE dans la ville et venant de différentes écoles françaises), je pense avoir beaucoup amélioré mon niveau d'anglais puisque je travaillais majoritairement en échangeant en anglais, et que mes deux colocataires étaient anglophones.

Le fait le plus marquant de mon stage a déjà été évoqué : il s'agit de la réorganisation majeure du groupe et le départ de tous les chefs d'équipes. J'ai en fait compris plus tard qu'ils avaient eu le choix de rester et de perdre leur statut de manager pour être au même niveau que tous les employés, ou bien de partir avec sans aucun doute une compensation financière intéressante ; ils ont bien entendu tous choisis la seconde option. Vivre une telle réorganisation d'entreprise est une expérience assez unique. Mais la façon dont cela s'est passé a été particulière.

Il y a en effet toujours eu un manque cruel de communication entre le VP et le CEO d'une part et le reste des employés d'autre part. Et cet événement en est l'exemple parfait. Aucune communication n'a été faite de la par du top management auprès des employés. Nous avons tous été mis au courant du départ de nos chefs d'équipes respectives de la même façon : au cours de la réunion quotidienne qui devait être une *news room,* les chefs d'équipes ont indiqué qu'ils avaient décidé de partir. C'est en discutant entre collègues d'équipes différentes que nous nous sommes rendu compte peu à peu qu'en réalité, tous les chefs d'équipes partaient le même jour et que cela n'était bien sûr pas une coincidence.

S'en sont suivis quelques jours où tout le monde était dans un état entre le questionnement et la peur car nous n'avons même pas reçu un mail pour expliquer la situation, ou ne serait-ce que nous indiquer à quel moment nous en saurions plus. Le CEO était bien entendu en déplacement en France lors du « départ » de ses managers, comme cela a toujours été le cas lors du départ d'un employé a-t-on appris ensuite.

Le climat était très particulier pendant ces quelques jours. Les spéculations concernant la suite logique allaient bon train, et de nombreux employés n'étaient pas serreins sur l'avenir du centre de San Francisco à ce moment précis. Pour certains employés français, leur visa est lié à la compagnie et s'ils ne travaillent plus pour elle, ils doivent quitter le territoire américain sous dix jours. On comprend leur inquiétude, d'autant plus lorsqu'ils sont installés depuis plusieurs années dans cette ville avec leur famille. Le manque de communication et la façon dont les choses se sont déroulées m'ont absolument choqué.

Les événements se sont soldés par l'annonce d'un séminaire au sud de San Francisco où tout le monde a été invité et où chacun devait présenter de nouveaux projets par équipe. Ce séminaire était en soit très sympatique à vivre car le cadre était agréable. Mais il s'est déroulé presque trois mois avant mon départ et aucun retour sur les propositions de projets présentées par tous les employés n'a été émis avant que je parte à ma connaissance. Tous les employés ont commencé à travailler sur leurs différents projets respectifs, mais l'ambiance générale avait changé.

Du point de vue de l'organisation du travail, l'un des faits les plus marquants de ce stage est la découverte d'un véritable modèle de management à l'américaine. Nous sommes jugés sur des livrables et des initiatives. Je pense que ce facteur est avant tout dû au fait que nous travaillions dans une logique exploratoire. Nous développons nos analyses le mieux possible, sur les tendances à venir, mais sans pouvoir garantir à 100% que nos prédictions se réaliseront telles quelles. Toute opposition est donc l'opportunité de revoir ses conclusions si besoin est, ou en tout cas d'adapter son discours et d'affiner son argumentation, ce qui est fort utile lorsque l'on s'adressera au top management des équipes françaises.

Le principal avantage de ce stage et de ce style de management est, selon moi, les opportunités offertes. En tant que poste avancé en charge d'identifier les innovations, nous avons (en théorie), l'oreille des top teams françaises. Tout dialogue se situe dès lors à un haut niveau. Comme je l'ai évoqué précédemment, nous sommes tout de suite mis en position de faire des démonstrations ou de rédiger des notes à destination du *board of directors*. La confiance accordée par le management du centre de San Francisco est centrale. Avoir la confiance de son chef motive à délivrer au moins le niveau des attentes. Une telle dynamique de résultats est très gratifiante. Je tiens d'ailleurs à souligner que dès le départ, on m'a fait confiance en me confiant d'importantes démonstrations et j'ai été beaucoup soutenu. J'ai toujours été intégré à des discussions impliquant des initiatives stratégiques importantes pour l'entreprise.

Ce modèle de management a toutefois ses défauts et ses qualités. Ainsi, comme il s'agit d'être très proactif, il n'y a pas vraiment de structure de travail bien définie.

Enfin, un autre fait important qui m'a marqué est la prise de conscience d'une difficulté majeure rencontrée par l'ensemble du centre de San Francisco et qui est la question des relations avec la France. Car depuis la fondation du centre de San Francisco, peu de projets ont été sélectionnés et ceux qui l'ont été n'ont pas rencontré le succès escompté. J'analyse cet état de fait de la façon suivante : le centre de San Francisco est dans une logique de start-up, c'est-à- dire analyser les évolutions des start-ups, essayer d'identifier des tendances et être les premiers à lancer des innovations.

FRANCE TELECOM, dans son ensemble est une très grande entreprise avec un cycle de décisions très différent des start-ups. Entre le moment où le centre de San Francisco identifie une opportunité, construit sa proposition et le moment où celle-ci est entendue, discutée puis acceptée en France, il peut s'être passé plusieurs mois, si ce n'est plusieurs années. Le moment opportun est passé depuis longtemps, alors qu'une start-up aura déjà développé la même idée et déjà été rachetée.

Une autre difficulté vient du fait qu'il est peu aisé de prouver à FRANCE TELECOM en quoi certaines innovations sont clés lorsque celles-ci sont encore récentes. Il nous est demandé sans cesse des *business cases* (des exemples d'entreprises similaires à elle ayant décidé de se lancer dans cette innovation). Alors que notre but n'est pas seulement de dire à FRANCE TELECOM quelles entreprises imiter. Car notre objectif est d'aider le groupe à se positionner en premier sur de nouveaux créneaux identifiés comme porteurs.

Et quand une de nos idées est jugée intéressante, les équipes en France redéveloppent souvent complètement le concept ce qui nécessite beaucoup de temps. Cela implique parfois qu'Orange rate l'opportunité de dévoiler un produit totalement innovant à temps.

En résumé, il n'est pas toujours simple pour OSV de faire admettre son importance en France.

3.2 Apports techniques

Ce stage m'a beaucoup apporté tant au niveau technique qu'au niveau de la culture d'entreprise. Le *Machine Learning* est un sujet qui me passionne sous tous les aspects : la théorie mathématique, la maîtrise informatique nécessaire d'outils puissants pour l'étude expérimentale, le monde de la recherche sur le sujet, les entreprises impliquées, sa mise en œuvre et les applications qui en découlent, mais également son avenir.

Techniquement j'ai pu approfondir mes connaissances dans de très nombreux domaines. Tout d'abord dans l'utilisation de Linux puisque j'ai travaillé pendant 6 mois sous cet environnement, chose qui ne m'était jamais arrivé pour une si longue période. Puis dans mes facultés de développement informatique puisque j'ai été confronté à de nombreux langages de programmation ; comme R, Java, Python, PHP ainsi que les outils relatifs à Hadoop. J'ai en effet suivi de nombreux tutoriels au début de mon stage pour me familiariser avec Hadoop, par exemple ceux concernant HortonWorks où j'ai découvert Pig, Hive, HBase et HDFS.

De plus, OSV dispose d'un Datacenter au sein de ses locaux afin de tester les solutions en interne. Cela m'a permis de disposer d'une capacité de calcul importante, mais également d'étendre mes connaissances en hardware en manipulant pour la première fois des serveurs et en apprenant comment monter les racks et les connecter physiquement au réseau. J'ai donc également beaucoup appris concernant les infrastructures informatiques à grande échelle, auxquelles on ne peut accéder lors de projets scolaires.

Comme déjà précisé, j'ai développé mes algorithmes de *Machine Learning* principalement sour R, j'ai alors découvert de nombreuses librairies intéressantes comme rmr2 ou rhdfs qui permettent de coder les tâches *map-reduce* directement sous R et d'utiliser ensuite une technologie appelée Hadoop Streaming qui permet justement de faire le lien entre Hadoop et des langages de programmation.

D'autres librairies sous R m'ont permis d'apprendre à paralléliser des calculs sur une seule machine en utilisant tous les cœurs disponibles comme les packages doparallel ou snowballC ; ce qui a été très utile étant donné que j'utilisais des machines avec en moyenne plus de 60 cœurs. Certaines librairies permettaient également de créer des clusters virtuels, ce qui accélérerait aussi les calculs.

J'ai aussi été initialisé à la virtualisation, notamment grâce à un ami stagiaire pour qui c'était la spécialisation et qui a travaillé sur différents projets (notamment autour de la technologie Docker) et qui m'a beaucoup appris sur le sujet. J'ai par exemple personnellement utilisé Vagrant qui est un outil de simulation d'un environnement de production permettant de créer des clusters avec des machines virtuelles en utilisant VirtualBox. Les avantages d'utiliser Vagrant résident par exemple justement lorsque l'on souhaite configurer un cluster, il n'y a alors aucune perte de temps pour configurer chacun des environnements. Il existe de simples fichiers de configuration et un ensemble de commandes permettant de définir le système et la configuration souhaitée pour son environnement. On peut ainsi déployer un gros *cluster* très rapidement pour faire des tests, sans avoir besoin d'avoir physiquement un tel *cluster*.

La virtualisation est en grande partie orientée backend, côté ingénieurs systèmes, développeurs et devops. L'une des applications majeures reste le déploiement de machines virtuelles sur un réseau afin de créer des clusters et des applications précises. C'est d'ailleurs dans cette direction que beaucoup de fournisseurs se sont orientés comme VMware, Citrix ou Microsoft qui fournissent en plus de la technologie d'hypervision les outils et services nécessaires pour gérer son cluster.

Que ce soient des outils de management locaux (VSphere, XenCenter…) ou accessibles via le web (Amazon Web Services, Google Cloud Platform), les entreprises visent à fournir des services performants de déploiement et de gestion afin de contrôler des infrastructures à large échelle. En plus de ces outils, la virtualisation est une technologie qui répond à ces problématiques en permettant le clonage de machines, le contrôle des ressources et du réseau. Ces options sont primordiales et peuvent êtres automatisées via des scripts informatiques, c'est ce que font certaines entreprises spécialisées comme Chef, Puppet ou encore des projets comme Vagrant.

Enfin, j'ai toujours pris un grand plaisir à tester toutes les dernières innovations technologiques, qu'il s'agisse de nouveaux produits physiques dans tous les domaines ou de nouvelles plateformes de traitement des données. Le dernier gadjet acheté et testé chez OSV avant que je parte était la dernière version d'Oculus Rift. Les graphismes ne sont pas encore saisissants mais la technologie a un avenir selon moi colossale.

- *D'ailleurs avec un tout petit peu de temps pour faire avaler la pilule morale, ce qui ne prendra que quelque générations, il paraît bien inévitable que l'on finisse par préfèrer voguer dans ces mondes virtuels qui seront bien plus acueillants et agréables que la réalité* - digression personnelle.

Du point de vue de la culture d'entreprise, j'ai beaucoup appris et j'ai pu comparer la culture américaine avec la culture française. Et les différences sont significatives, d'autant plus étant dans la Silicon Valley. L'expérience à San Francisco a été totalement différente de mes stages précédents réalisés en France. Le travail en Openspace, les timesheets ou encore les newsroom journalières, tout ceci forge une expérience originale et propre aux Etats-Unis. De plus, travailler dans une entreprise dans la Silicon Valley est un grand plus, on est proche de toutes les start-ups qui deviendront peut être un jour de grandes firmes mais également proche des grands groupes comme Google, Facebook, Twitter, Apple, ou Cisco. Il est aisé de rencontrer des entreprises innovantes et les meetings avec ces entreprises font partie intégrante du travail et de la culture de la *Silicon Valley*.

3.3 Apports personnels

Mais ce que m'aura apporté ce stage dans cet environnement extraordinaire ne saurait se résumer si simplement. Aussi je tenterai ci-après de parcourir quelques uns des points notables qui m'auront énormément apportés personnellement.

La vie à San Francisco

Il n'est pas de stage à l'étranger sans parler un peu de l'environnement dans lequel on évolue et particulièrement la ville dans laquelle on travaille. Dans mon cas il s'agissait donc de San Francisco, ville aux multiples facettes située en Californie et dont je suis peu à peu tombé amoureux. Voici dans la suite ce qui m'aura le plus marqué.

Le Brouillard

L'association du courant froid et de la chaleur de la Californie intérieure est responsable des nappes de brouillard caractéristiques qui se forment dans certains quartiers de la ville et au-dessus des eaux de la Baie pendant l'été et au début de l'automne. De ce fait, les températures estivales à San Francisco sont généralement beaucoup plus basses que dans d'autres endroits de la Californie Cette fraicheur estivale est sans doute à l'origine d'une légende urbaine selon laquelle Mark Twain aurait écrit *« The coldest winter I ever spent was a summer in San Francisco ».*

Cela m'a d'ailleurs toujours impressionné lorsqu'en rentrant de weekends dans différents endroits de Californie, il était possible, en quelques kilométres d'autoroute seulement, de perdre parfois 20 degrés en rentrant dans la baie.

Les Quartiers

Beaucoup de grandes villes aux Etats-Unis possèdent des quartiers bien distincts et San Francisco n'échappent pas à la règle. Passer d'un quartier à l'autre se ressent parfois comme passer d'un pays à l'autre, tant du point de vue architectural que culturel, ce qui est tout à fait fascinant et l'expérience vaut le détour.

On retrouve les classiques Financial District, Chinatown et Little Italy mais également d'autres quartiers qui ont forgés la réputation de San Francisco comme Castro (quartier Gay), Mission (quartier Latino) ou encore Haight Valley avec ses maisons victoriennes traditionnelles et la vue sur la ville à Alamo Square. Avec sa proximité avec le Golden Gate Park et Haight Street, la rue hippie, ainsi qu'avec ses maisons en bois coloré magnifiques, ce quartier a d'ailleurs dès le début été mon préféré. J'ai eu la chance d'y résider en collocation avec des américains, à deux blocs seulement d'Alamo Square.

C'est aussi à San Francisco qu'on peut trouver la fameuse *Maison Bleue* chantée par Maxime Le Forestier, au pied du très agréable *Delores Park* où les gens de tous horizons aiment se mêler. D'autres quartiers sont également réputés comme *Pacific Height* et ses luxueuses demeures, la *Marina* et son *Pier 39* ou encore *Japan Town* et ses commerces Japonais. J'ai alors été surpris mais subjugué par une diversité culinaire impressionante et selon moi bien plus développée qu'à Paris, pour mon plus grand bonheur.

La Silicon Valley

Cette carte résume plutôt bien visuellement ce que représente la *Silicon Valley* au monde High-Tech : les innovations technologiques viennent en grande partie d'ici car la communauté *tech* tout comme la façon de travailler y sont uniques.

Une partie des applications sont créées ici et sont directement utilisées par tout le monde. Certaine rentrent rapidement dans le quotidien des gens comme elles l'ont fait pour moi. Je n'en citerai qu'une parmi tant, Venmo, qui est comme une banque en ligne instantanée mixée à un réseau social et qui facilite le remboursement d'argent dès qu'une activité est faite à plusieurs. Autant dire que cette application m'a servie quasiment quotidiennement.

Les Ponts et collines

Plusieurs ponts relient la ville aux rives de la baie: les plus célèbres sont le Golden Gate Bridge (au nord-ouest) et le Bay Bridge, qui relie San Francisco à Oakland vers l'est et sur lequel nous avions une superbe vue depuis nos bureaux chez OSV.

San Francisco est célèbre pour les plus de 50 collines situées à l'intérieur des limites de la ville. Certaines d'entre elles correspondent à un quartier, comme Nob Hill, Pacific Heights, Russian Hill ou Telegraph Hill; d'autres sont des jardins publics ou des parcs comme *Buena Vista* ou *Twin Peaks* d'où la vue sur la ville est magnifique. Se promener en voiture dans la ville à travers ses collines aux rues parfois très pentues restera un souvenir fort. Un point qui m'a toujours frappé est qu'une rue peut être très raide en montée alors que la première rue paralléle à celle-ci, sur le même block donc, peut être raide mais cette fois en descente. Autant dire que l'on retient vite la géographie de la ville lorsque l'on s'y déplace en vélo car une petite erreur d'itinéraire peut vite être douloureuse, ou au contraire un détour peut s'avérer être un gain de temps important s'il emprunte davantage de rues en descente.

Les Transports

Ce qui m'a également surpris est l'archaïsme des transports en commun au cœur même des innovations technologique, comparé à Paris tout au moins. Il y a très peu de lignes de métro et elles ne desservent pas toute la ville. Quant aux bus, il sont vraiment anciens et assez peu fréquents. C'est sans doute pour cela que les taxis sont très utilisés dans cette ville.

UBER

Mais parce que prendre le taxi est aussi courant que prendre son café au Peet's ou au Starbucks le matin et que l'on est dans la Silicon Valley, des services de taxis indépendants ont étés créés, avec bien évidement des applications mobiles pour couronner le tout. Ces applications, comme *Lyft* ou *Uber*, permettent de commander via son Smartphone un taxi privé le plus proche possible de l'endroit où l'on se trouve et ceci n'importe quand. Les courses sont souvent moins chères que des taxis normaux et la disponibilité de ces taxis privés est bien plus grande que les classiques. Et comme d'habitude, la bonne humeur des conducteurs est toujours au rendez-vous. Il va être vraiment difficile de s'en passer de retour en France.

Les Rencontres

L'un des éléments les plus frappants que j'ai pu observer à San Francisco restera sans aucun doute les rencontres qu'il est possible de faire à tout moment. Les Californiens sont très sociables et engagent la conversation très facilement.

C'est en fait l'état d'esprit général des gens que j'ai trouvé totalement différent de celui qu'on peut trouver en France et en particulier à Paris. En effet, dans la Silicon Valley, personne ne porte de jugements attifs, et il y a une espèce de culture de l'écoute et de l'entraide impressionante : les carnets d'adresse s'échangent en un rien de temps, et pas besoin d'avoir fait ses preuves auparavant.

D'ailleurs la culture de l'échec est complétement différente également. Les échecs sont perçus comme très formateurs, enrichissants voire nécessaires. Dans la Silicon Valley, presque tout le monde est entrepreneur. On est bien loin des mentalités superficielles, arrogantes et prétencieuses qu'on trouve partout à Paris ; je dirais même qu'on est à l'exact opposé.

L'atmosphère générale est détendue, authentique et agréable. Cela se ressent même sur la façon de se vétir des gens, aucun ingénieur de la *tech* ne travaille en costume par exemple, la seule fois où je vais en porter un en plus de 6 mois de temps sera pour ma soutenance de stage, et cela ne m'a guère manqué. Le mythe des Californiens travaillant dans des start-up en short et en tong est en fait véridique. De façon générale, il n'y a aucun jugement porté sur l'apparence des gens.

Les gens ici aiment beaucoup se réunir après le travail pour discuter d'un sujet donné, pour faire des rencontres ou organiser des *hackatons*. Là encore une application très populaire a été créée spécialement pour rencontrer des personnes autour de thèmes et affinités communes : *Meetup*. C'est grâce à cette application que j'ai pu d'abord visiter de nombreux *Open-Space*, tous plus impressionnants les uns que les autres par le bien être qu'on s'efforce à y faire régner ; mais cela m'a aussi permis d'échanger avec de nombreux ingénieurs des plus grosses entreprises du moment, autour d'une bière et d'un plateau de fromages.

Conclusions et Perspectives futures

Mon stage ne peut pas se résumer simplement à une découverte du monde du travail, car ce fut également l'occasion de découvrir un nouveau pays et une nouvelle culture.

Tout d'abord, ce stage fut ma première réelle expérience professionnelle et aussi une agréable surprise. J'ai eu la chance d'intégrer un environnement de travail qui me convenait au sein d'une équipe dynamique et motivée. Celle-ci m'a donné la chance de travailler sur des projets à la fois passionnants et instructifs. Je pense donc que cette expérience complète parfaitement ma formation, avec en plus un perfectionnement de la langue anglaise et une solide connaissance des enjeux technologiques de demain.

Mais, cette expérience n'aurait pas été un succès et aussi agréable si elle n'avait pas été accompagnée par des découvertes faites lors de mes nombreux voyages durant mes weekends. J'ai ainsi pu découvrir la Californie, et j'ai d'ailleurs remarqué que le mode de vie des américains n'est finalement pas si lointain des clichés de l'industrie cinématographique américaine !

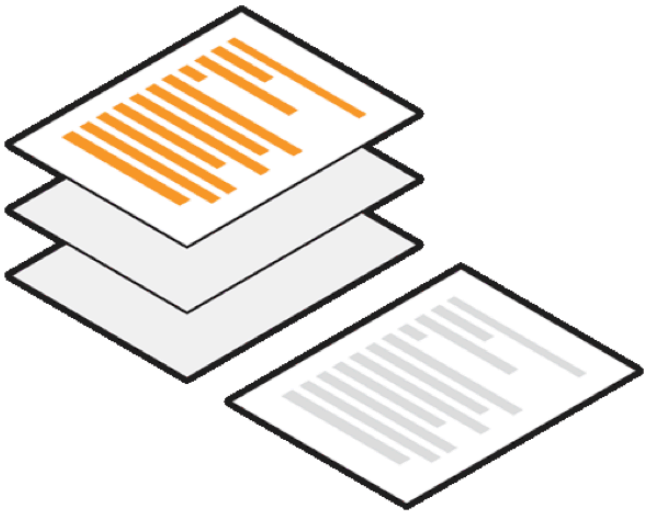
J'ai également eu la chance de gouter à la vie d'expatrié aux Etats Unis qui est très plaisante, d'autant qu'il y a de nombreux français dans la Silicon Valley. Il est toujours très agréable de se retrouver entre français pour partager nos expériences et échanger quelques petites astuces de la vie quotidienne.

J'aurais sans aucun doute oublié de mentioner de très nombreux faits marquants, je me suis restreint aux quelques uns me venant directement à l'esprit lors de la rédaction de ce rapport. Je n'ai par exemple pas encore mentionné la *wellness room* qui était une pièce chez OSV destinée à la détente. N'importe quel employé pouvait demander les clés de cette pièce singulière, à n'importe quel moment de la journée, et profiter du canapé, des coussins et des couvertures pour faire une courte sieste.

J'ai aussi fait le choix de ne pas alourdir ce rapport avec des pages de code. En revanche, je posterai mon travail sur mon futur GitHub qu'il faut que je crée depuis longtemps déjà.

Les progrés que j'ai fait pendant ce stage sont davantage techniques que théoriques, mais cela n'est pas un soucis car je suis excité et ravi d'avoir été accepté l'an prochain dans le Master 2 MVA (Mathèmatiques, Vivion et Apprentissage) que propose l'ENS de Cachan. Je vais donc vraissemblablement beaucoup progresser théoriquement et mettre à profit mes nouvelles connaissances techniques.

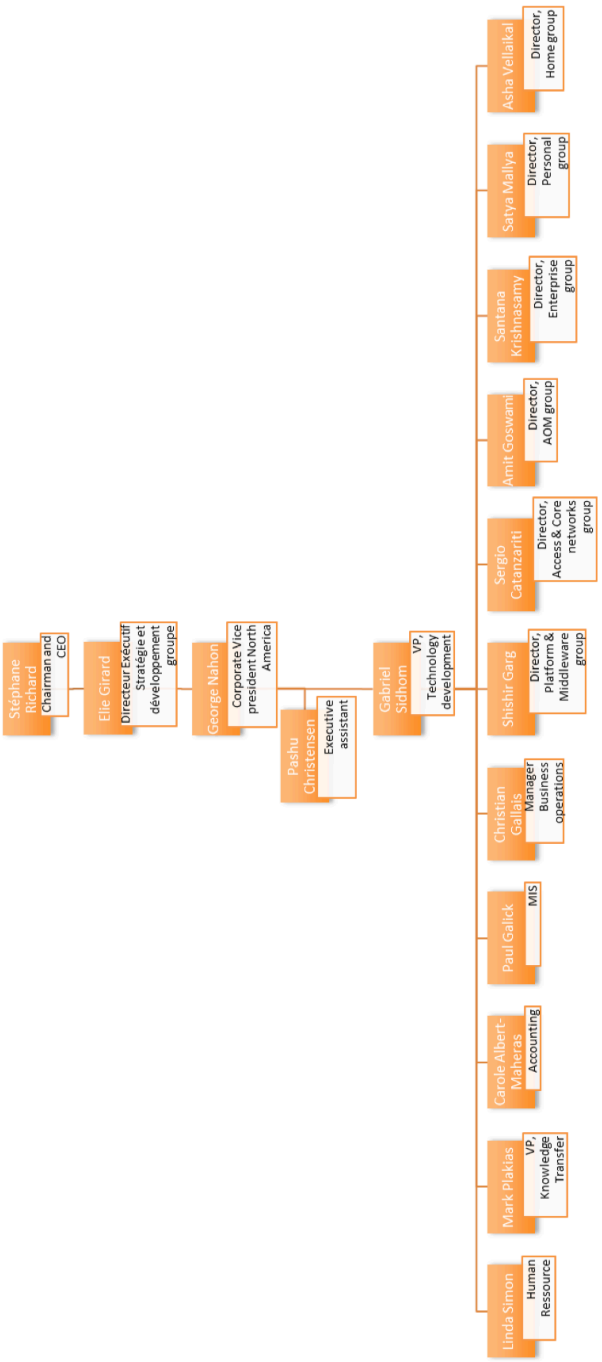
J'ai tout à fait conscience que l'expérience Californienne que j'ai vécu est une chance extraordinaire pour un ingénieur et effectuer un stage de fin d'études dans le cadre que j'ai tenté de décrire tout au long de ce rapport m'a apporté énormément. J'espère que j'aurais réussi à transmettre dans ce rapport la joie qui a été la mienne pendant ces six mois qui resteront un souvenir inoubliable dans ma vie.



Annexes

I. Organigramme d'Orange Silicon valley

Il s'agit de l'organisgrame tel qu'il était pendant la première partie de mon stage, avant la réorganisation et le départ des chefs d'équipes.



II. Unicorns

Voici les *unicorns* utilisées pour mon projet, classées par catégories et tirées d'un rapport édité pendant mon stage par des collègues d'OSV (voire bibliographie).

Big Data :

*as of April 10, 2014

unicorn	valuation $B	year founded	status	acquirer
Climate Corp	$.93B	2006	Corporate	Monsanto
Palantir	$9B	2004	Private	
MongoDB	$1.2B	2007	Private	
Splunk	$7.7B	2003	Public	
Cloudera	$1.8B	2008	Private	
Hortonworks	$1.6B	2011	Private	
Tableau Software	$4.06B	2003	Public	
Marketo	$1.22B	2006	Public	
RocketFuel	$1.38B	2008	Public	
Veeva Systems	$2.84B	2007	Public	
Waze	$1.1B	2007	Corporate	Google
Nest	$3.2B	2010	Corporate	Google
Nutanix	$1B	2009	Private	
Nimble Storage	$2.34B	2008	Public	

Palantir splunk> +ableau SOFTWARE Veeva nest

Cloud :

*as of April 10, 2014

unicorn	valuation $B	year founded	status	acquirer
Box	$3B	2005	Private	
Dropbox	$10B	2007	Private	
Actifio	$1.1B	2009	Private	
Atlassian	$3.3B	2002	Private	
Mandiant	$1.1B	2004	Corporate	FireEye
Palo Alto Networks	$4.88B	2005	Public	
FireEye	$7.12B	2004	Public	
Evernote	$1B	2004	Private	
Meraki	$1.2B	2006	Corporate	Cisco
Niciri	$1.25B	2007	Corporate	VMWare

box Dropbox ATLASSIAN paloalto NETWORKS FireEye

E-Commerce :

unicorn	valuation $B	year founded	status	acquirer
Gilt	$1.1B	2007	Private	
Fab.com	$1.1B	2011	Private	
Yelp	$4.82B	2004	Public	
RetailMeNot	$1.76B	2007	Public	
Groupon	$4.93B	2008	Public	
Zulily	$5.78B	2010	Public	
Lending Club	$1.5B	2006	Private	
Stripe	$1.75B	2010	Private	
Square	$5B	2009	Private	
Airbnb	$10B	2008	Private	
Kayak	$1.8B	2004	Corporate	Priceline
Lyft	$1B	2007	Private	
Uber	$3.5B	2009	Private	
Eventbrite	$1.1B	2006	Private	
Homeaway	$3.12B	2005	Public	

Entreprise :

*as of April 10, 2014

unicorn	valuation $B	year founded	status	acquirer
LinkedIn	$20.47B	2003	Public	
Workday	$14B	2005	Public	
Yammer	$1.2B	2008	Corporate	Microsoft
Pure Storage	$1.1B	2009	Private	
Fusion-io	$1.04B	2006	Public	
ServiceNow	$7.23B	2004	Public	
DocuSign	$1.6B	2003	Private	

Mobile :

*as of April 10, 2014

unicorn	valuation $B	year founded	status	acquirer
Instagram	$1B	2010	Corporate	Facebook
Airwatch	$1.2B	2003	Corporate	VMWARE
WhatsApp	$19B	2009	Corporate	Facebook
SnapChat	$2B	2011	Private	

Media :

*as of April 10, 2014

unicorn	valuation $B	year founded	status	acquirer
Zynga	$3.65B	2007	Public	
Oculus VR	$2B	2012	Corporate	Facebook
Hulu	$2B	2007	Private	
GoPro	$2.5B	2003	Private	
Youtube	$1.65B	2005	Corporate	Google

Social Networking :

*as of April 10, 2014

unicorn	valuation $B	year founded	status	acquirer
Tumblr	$1.1B	2007	Corporate	Yahoo
Pinterest	$3.8B	2009	Private	
Tango	$1.1B	2009	Private	
Twitter	$23.2B	2006	Public	

III. TFIDF

This technique, known as Term Frequency - Inverse Document Frequency or simply TF-IDF, weights a given term to determine how well the term describes an individual document within a corpus. It does this by weighting the term positively for the number of times the term occurs within the specific document, while also weighting the term negatively relative to the number of documents which contain the term. Consider term t and document $\in D$, where t appears in n of N documents in D. The TF-IDF function is of the form:

$$\text{TFIDF}(t,d,n,N) = \text{TF}(t,d) \times \text{IDF}(n,N)$$

There are many possible TF and IDF functions. I've made my choice for this one :

$$\text{TF}(t,d) = \sum_{word \in d} 1(if\ t \in d)$$

Additionally, the term frequency may be normalized to some range, what I have done with the number of article for a given time slot. This is then combined with the IDF function. The IDF function I used is this one:

$$\text{IDF}(n,N) = \log\left(\frac{N-n}{n}\right)$$

When the TF-IDF function is run against all terms in all documents in the document corpus, the words can be ranked by their scores. A higher TF-IDF score indicates that a word is both important to the document, as well as relatively uncommon across the document corpus. This is often interpreted to mean that the word is significant to the document, and could be used to accurately summarize the document.

TF-IDF provides a good heuristic for determining likely candidate keywords, and it (as well as various modifications of it) have been shown to be effective after several decades of research. Several different methods of keyword extraction have been developed since TF-IDF was first published in 1972, and many of these newer methods still rely on some of the same theoretic backing as TF-IDF. Due to its effectiveness and simplicity, it remains in common use today.

IV. Time series construction and Machine Learning Design

A time series is a sequence S of historical measurements y_t of an observable variable y at equal time intervals. An important aspect of the forecasting task is represented by the size of the horizon. If the one-step forecasting of a time series is already a challenging task, performing multi-step forecasting is more difficult because of additional complications, like accumulation of errors, reduced accuracy, and increased uncertainty.

The first step in the construction is to extract 3 keywords from every article from DaVinci - the words considered as the most important by a text mining algorithm. It could be a domain or a company name for example.
The next step is to select from all this keywords, those appearing at least in 50 (for example) different articles to obtain the names of our time series.
The final step is to associate a 290*1 vector for each keyword, with the first component = the number of articles related to this keyword and posted the last week of 2013; and the 290th component = the number of articles related to this keyword and posted the first week of 2008 (from 2008 to 2014, there are 6 years with 48 weeks each, so 6*48=288 + 2 weeks to be precise).

So this first idea is to give a score evolution that is only based on the frequency of keywords occurrence, nevertheless it could prove that the DaVinci data set is really rich and could be used for prediction of the high tech market trends.

In the following, we will analyze our time series within the framework of a dynamical systems approach. Therefore the time series are interpreted as the observable of a dynamical system whose state s evolves in a state space $\Gamma \subset \mathbb{R}^g$, according to the law:

$$s(t) = F^t(s(0))$$

Where $F : \Gamma \to \Gamma$ is the map representing the dynamics, F^t is its iterated version and $s(t) \in \Gamma$ denotes the value of the state at time t.
In the absence of noise the time series is related to the dynamical system by the relation:

$$y_t = G(s(t))$$

Where $G : \Gamma \to \mathbb{R}^D$ is called the *measurement function* and D is the dimension of the series. In our case of study, D = 1 because we have univariate time series with scores that we have constructed evolving in a subset of $\mathbb{R}$.

Both the functions F and G are unknown, so in general we can't hope to reconstruct the state in its original form. However, we may be able to recreate a state space that is in some sense equivalent to the original.
The state *space reconstruction problem* consists in reconstructing the state when the only available information is contained in the sequence of observations y_t , which is precisely our problem.

Different theorems implies that for a wide class of systems, a smooth dynamics $f: \mathbb{R}^n \to \mathbb{R}$ is induced in the space of reconstructed vectors:

$$y_t = f(y_{t-d}, y_{t-d-1}, \dots, y_{t-d-n+1})$$

Where *d* is called the *lag time* and *n* (*order*) is the number of past values taken into consideration.
This implies that the reconstructed states can be used to estimate *f* and consequently *f* can be used in alternative to F and G, for any purpose concerning time series analysis, including forecasting.

The last representation does not take into account any noise component, since it assumes that a deterministic process f can accurately describe the time series.
But once we assume that we have not access to an accurate model of the function *f*, it is perfectly reasonable to extend the last deterministic formulation to a statistical Nonlinear Auto Regressive (NAR) formulation:

$$y_t = f(y_{t-d}, y_{t-d-1}, \dots, y_{t-d-n+1}) + w(t)$$

Where the missing information is lumped into a noise term *w*. We will then refer to this formulation as a general representation of our time series.
The success of a reconstruction approach starting from a set of observed data depends on the choice of the hypothesis that approximates *f*, the choice of the order *n* and the lag time *d*. We will take in the following *d*=0 and we will have to test different order values.

- **Preprocessing Methods**

The time series constructed will have a variety of features. Some will possess seasonality, some will exhibit a trend (exponential or linear), and some will be trendless, just fluctuating around some level. Some preprocessing needs to be done to handle these features.

A focus of different studies has been to examine preprocessing methods, used in conjunction with the machine learning forecasting models, such as deseasonalization, taking the log transformation, scaling, differencing the time series, or taking moving averages. The scaling step is essential to get the time series in a suitable range, especially for the Multilayer Perceptron (MLP) algorithm where scaling is necessary. We can use linear scaling computed using the training set, to scale the time series to be between -1 and 1.

These methods have shown that they have a big impact on the subsequent forecasting performance. It is as if we are "making it easier" for the forecasting model by transforming the time series based on information we have on some of its features. For example, a large number of empirical studies advocate the deseasonalization of data possessing seasonalities for neural networks.
Other preprocessing methods such as taking a log transformation and detrending have also been studied in the literature. We will try a deseasonalization step (if needed) and a log-step, then other preprocessings used for machine learning forecasting models and having good results:

- No special preprocessing (LAGGED-VAL): the input variables to the machine learning model are the lagged time series values, say $(y_{t-n+1}, \dots, y_t)$ and the value to be predicted (target output) is the next value (for one-step ahead forecasting).

- Taking moving averages (MOV-AVG): We compute moving averages with different sized smoothing windows, for example:

$$u_i(t) = \frac{1}{J_i} \sum_{j=t-J_i+1}^{t} y_j \text{ , for } i = 1, \dots, I$$

where J_i is the size of the averaging window. The new input variables for the forecasting model would then be $u_i(t)$ and the target output is still y_{t+1}. The possible advantage of this preprocessing method is that moving averages smooth out the noise in the series, allowing the forecasting model to focus on the global properties of the time series. The existence of several moving averages with different smoothing levels is important so as to preserve different levels of time series detail in the set of inputs.

After these transformations are performed we extract the input variables from the transformed time series. Then the forecasting model is applied. Once we perform the forecasting, we unwind all these transformations of course in reverse order.

We will use as error measure the symmetric mean absolute percentage error, defined as:

$$SMAPE = \frac{1}{M} \sum_{m=1}^{M} \frac{|ye_m - y_m|}{(|ye_m| + |y_m|)\backslash 2}$$

where y_m is the target output, ye_m is the prediction and M the number of time series in the training set. Since it is a relative error measure it is possible to combine the errors for the different time series into one number.

- **Parameters determination**

For every considered method there are typically a number of parameters, some of them are key parameters and have to be determined with care. The key parameters are the ones that control the complexity of the model: for example the order *n*, the size of the network for MLP etc.

For linear models, the typical approach for model selection is to use an information criterion such as Akaike's criterion, the Bayesian information criterion, or others, which consider a criterion consisting of the estimation error added to it a term penalizing model complexity. For machine learning approaches such criteria are not well-developed yet. Even though some theoretical analyses have obtained some formulas relating expected prediction error with the estimation error (training error) and model complexity, these formulas are mostly bounds and have not been tested enough to gain acceptance in practical applications. The dominant approach in the machine learning literature has been to use the K-fold validation approach for model selection. Empirical comparisons indicate its superiority for performance accuracy estimation and for model selection over other procedures such as the hold out, the leave one out and the bootstrap methods.

In the K-fold validation approach the training set is divided into K equal parts (or folds). We train our model using the data in the K−1 folds and validate on the remaining K^{th} fold. Then we rotate the validation fold and repeat with the same procedure again. We perform this training and validation K times, and compute the sum of the validation error obtained in the K experiments.

This will be the validation error that will be used as a criterion for selecting the key parameters. For each method there are generally two parameters (or more) that have to be determined using the K-fold validation: the number of input variables (i.e. the order *n*), and the parameter determining the complexity (for example the number of hidden nodes for MLP, say NH in the following).

- **Overview of ML techniques in time series forecasting**

Here is a short presentation of different manner to process during the ML phase. The idea is to build the model with a subset of our final time series data set using R, and

then to stream the algorithms on Hadoop for parallel mining of the entire table and for potential huge amount of data if we want to try to make trend forecasting with others datasets available in the web.

1. Formalization of one-step forecasting problems as supervised learning task

The embedding formulation we proposed in the second section suggests that, once a historical record S is available, the problem of one-step forecasting can be tackled as a problem of supervised learning. Supervised learning consists in modeling, on the basis of a finite set of observations $\{(X_i, Y_i)_{\ i\in\{1,\dots,N\}}\}$, the relation between a set of input variables $\{X_i\}$ and one or more output variables $\{Y_i\}$, which are considered somewhat dependent on the inputs. In one-step forecasting, the *n* previous values of the series are available and the forecasting problem can be cast in the form of a generic regression.

To be more explicit, let's take an example. If we have the time series $(y_1, y_2, \dots, y_t)$ and we want to predict y_{t+1}, the training set of the machine learning algorithms would be for *t* =6 and *n* = 3 :

$$\{ (X_i, Y_i) \} = \{ ((y_1, y_2, y_3) , y_4) , ((y_2, y_3, y_4) , y_5) , ((y_3, y_4, y_5) , y_6) \}$$

So here X_i is a vector and Y_i a scalar.

2. ML algorithms choices

Large scale comparison studies of a variety of machine learning models applied to business-type time series forecasting (such as the M3 competition data) have been conducted for classification and regression problems.
In the previous section, we modeled the problem as a regression one, since our scores take its values in $\mathbb{R}$.

For a regression problem, the following models have been compared: multilayer perceptron (MLP), Bayesian neural networks, radial basis functions, generalized regression neural networks, K-nearest neighbor regression, CART regression trees, support vector regression, and Gaussian processes.
The conclusions of these studies show that the model the best ranked is multilayer perceptron, or often simply called neural network. So I will be concentrated above all on this model.

To be a bit more precise, let's explicit the multilayer perceptron with only one hidden layer model mainly studied:

This model is perhaps the most popular network architecture in use today both for classification and regression. The MLP is given as follows:

$$y = v_0 + \sum_{j=1}^{NH} v_j g(w_j^T x')$$

where x is the input vector and $x' = (1, x^T)^T$, w_j is the weight vector for the j^{th} hidden node, v_0 , v_1 , … , v_{NH} are the weights for the output node, and y is the network output. The function g represents the hidden node output, and it is given in terms of a squashing function, for example the logistic function:

$$g(u) = \frac{1}{1+\exp(-u)}$$

The MLP is a heavily parametrized model, and by selecting the number of hidden nodes NH we can control the complexity of the model. The breakthrough that lent credence to the capability of neural networks is the universal approximation property. Under certain mild conditions on the hidden node functions g, any given continuous function on a compact set can be approximated as close as arbitrarily given using a network with a finite number of hidden nodes. While this is a reassuring result, it is critical to avoid overparametrization, especially in forecasting applications which typically have a limited amount of highly noisy data.

Model selection (via selecting the number of hidden nodes) has therefore attracted much interest in the neural networks literature.
We can for example use a K-fold validation procedure to select the number of hidden nodes. To obtain the weights, the mean square error is defined and the weights are optimized using gradient techniques. The most well-known method, based on the steepest descent concept, is the backpropagation algorithm. A second order optimization method called Levenberg Marquardt is generally known to be more efficient than the basic backpropagation algorithm.

In other words, it is a multilayer Perceptron but only with one hidden layer. Thanks to the great breakthrough in deep learning and neural networks using several hidden layers, we could try to apply these theories and see if prediction accuracy is better or not.

3. From one-step to multi-step forecasting

The previous sections showed that one-step forecasting can be cast in a conventional supervised learning framework, and the aim is to extend it to show how learning techniques can be used to tackle the multi-step forecasting problem. Indeed we want to predict not only the next score of a domain but his middle term evolution.
A multi-step time series forecasting task consists of predicting the next H values $(y_{N+1}, \dots, y_{N+H})$ of a historical time series $(y_1, \dots, y_N)$ composed of N observations, where $H > 1$ denotes the forecasting horizon.
We will use a common notation where f and F denote the functional dependency between past and future observations, n refers to the embedding dimension of the time series, that is the number of past values used to predict future values and w represents the term that includes modeling error, disturbances and/or noise.

So there exist different strategies to adopt machine learning in multi-step forecasting. A first strategy to be considered is the recursive one.
These strategy trains first a one-step model f :

$$y_{t+1} = f(y_t, \dots, y_{t-n+1}) + w(t+1)$$

with $t \in \{n, \dots, N-1\}$ and then uses it recursively for returning a multi-step prediction. An obvious drawback of the recursive method is its sensitivity to the estimation error, since estimated values are more and more used when we get further in the future.

Next we have the direct strategy which learns independently H models f_h:

$$y_{t+h} = f_h(y_t, \dots, y_{t-n+1}) + w(t+h)$$

with $t \in \{n, \dots, N-H\}$ and $h \in \{1, \dots, H\}$ and returns a multi-step forecast by concatenating the H predictions.
So it is a really simple strategy which moreover doesn't use any approximated values to compute the forecasts, so there is no errors accumulation. Nevertheless, it has some weaknesses: first, since the H models are learned independently, no statistical dependencies between the predictions are considered. Then, this strategy demands a large computational time since the number of models to learn is equal to H.

Then, the DirRec strategy combines the architectures and the principles of the two previous strategies. DirRec computes the forecasts with different models for every horizon (like the direct strategy) and, at each time step, it enlarges the set of inputs by adding variables corresponding to the forecasts of the previous step (like the recursive strategy). However the embedding size n is different for all the horizons. In other terms, the DirRec strategy learns H models f_h from the time series $(y_1, \dots, y_N)$ where:

$$y_{t+h} = f_h\,(y_{t+h-1}\,, \dots\,, y_{t-n+1}) + w(t+h)$$

with $t \in \{n, \dots, N-H\}$ and $h \in \{1, \dots, H\}$. So a very computational time consuming strategy.

In spite of their diversity, iterated and direct techniques for multiple-step forecasting share a common feature: they model from data a multi-input single-output mapping whose output is the variable y_{t+1} in the iterated case and the variable y_{t+k} in the direct case, respectively. When a very long term prediction is at stake and a stochastic setting is assumed, the modeling of a single-output mapping neglects the existence of stochastic dependencies between future values, (e.g. between y_{t+k} and y_{t+k+1}) and consequently biases the prediction accuracy. As we would like to predict long term scores (some months or one year), this is a problem for us.
A possible way to remedy to this shortcoming is to move from the modeling of single-output mappings to the modeling of multi-output dependencies. This requires the adoption of multi-output techniques where the predicted value is no more a scalar quantity but a vector of future values of the time series.

The Multi-Input Multi-Output (MIMO) strategy avoids the simplistic assumption of conditional independence between future values made by the direct strategy by learning a single multiple-output model:

$$(y_{t+H}\,, \dots\,, y_{t+1}) = F(y_t\,, \dots\,, y_{t-n+1}) + w$$

where $t \in \{n, \dots, N-H\}$, $F : \mathbb{R}^n \rightarrow \mathbb{R}^H$ is a vector-valued function, and $w \in \mathbb{R}^H$ is a noise vector with a covariance that is not necessarily diagonal.
So the forecasts are returned in one step by a multiple-output model and the MIMO strategy model the stochastic dependency characterizing the time series. This strategy avoids the conditional independence assumption made by the direct strategy as well as the accumulation of errors which plagues the recursive strategy.

Finally, the DIRMO Strategy aims to preserve the most appealing aspects of DirRec and MIMO strategies by partitioning the horizon H in several blocks, and using MIMO to forecast the values inside each block. This means that the H -step forecast requires m multiple-output forecasting tasks ($m = \frac{H}{s}$), each having an output of size s ($s \in \{1, \dots, H\}$).
In fact, for $s = 1$, the DIRMO coincides with the conventional direct strategy, while for $s = H$ it corresponds to the MIMO strategy. The tuning of the parameter s allows to improve the flexibility of the MIMO strategy by calibrating the dimensionality of the outputs (no dependency in the case $s = 1$ and maximal dependency for $s = H$). This provides a beneficial tradeoff between preserving a larger degree of the stochastic dependency between future values and having a greater flexibility of the predictor.

Liste de références

Xavier Quintuna
Software and IT Architect in Big Data
Orange Silicon Valley
60 Spear Street
Suite 11 San Francisco, CA 94105
xavier.quintuna@orange.com
xavierqa@gmail.com
+1 650 438 6568

Shishir Garg
Director of Middleware
Orange Silicon Valley
60 Spear Street
Suite 11 San Francisco, CA 94105
shishir.garg@orange.com
+1 415 243 1540

Bibliographie

G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. Science, 28 July 2006, Vol 313.

Geoffrey E. Hinton. Learning multiple layers of representation. ScienceDirect, Vol.11 No.10.

Yaniv Taigman, Ming Yang and Marc'Aurelio Ranzato. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. Facebook AI Group.

Conrad S. Tucker and Harrison M. Kim. Trend Mining for Predictive Product Design. JournalofMechanicalDesign 025108JMD

Gianluca Bontempi, Souhaib Ben Taieb and Yann-Aël Le Borgne. Machine Learning Strategies for Time Series Forecasting. Technical report, 2013.

Olga Streibel. Mining Trends in Texts on the Web. Technical report, 2008.

Nesreen K. Ahmed and Amir F. Atiya. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. Technical report, 2010.

Souhaib Ben Taieba, Gianluca Bontempia, Amir Atiyac and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. arXiv:1108.3259v1 [stat.ML] 16 Aug 2011.

Juan Ramos. Using TF-IDF to Determine Word Relevance in Document Queries. Technical report, 2003.

Brian Lott. Survey of Keyword Extraction Techniques. Technical report, December 2012.

Sayantani Ghosh, Sudipta Roy, and Samir K. Bandyopadhyay. A tutorial review on Text Mining Algorithms. International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 4, June 2012.

Naresh Kumar Nagwani. Clustering Based URL Normalization Technique for Web Mining. International Conference on Advances in Computer Engineering, 2010.

Jordan Bates, Jennifer Neville and Jim Tyler. Using Latent Communication Styles to Predict Individual Characteristics. Technichal report, 2012.

Rapport Unicorn : http://fr.slideshare.net/orangesv/unicorns-startups-and-giants