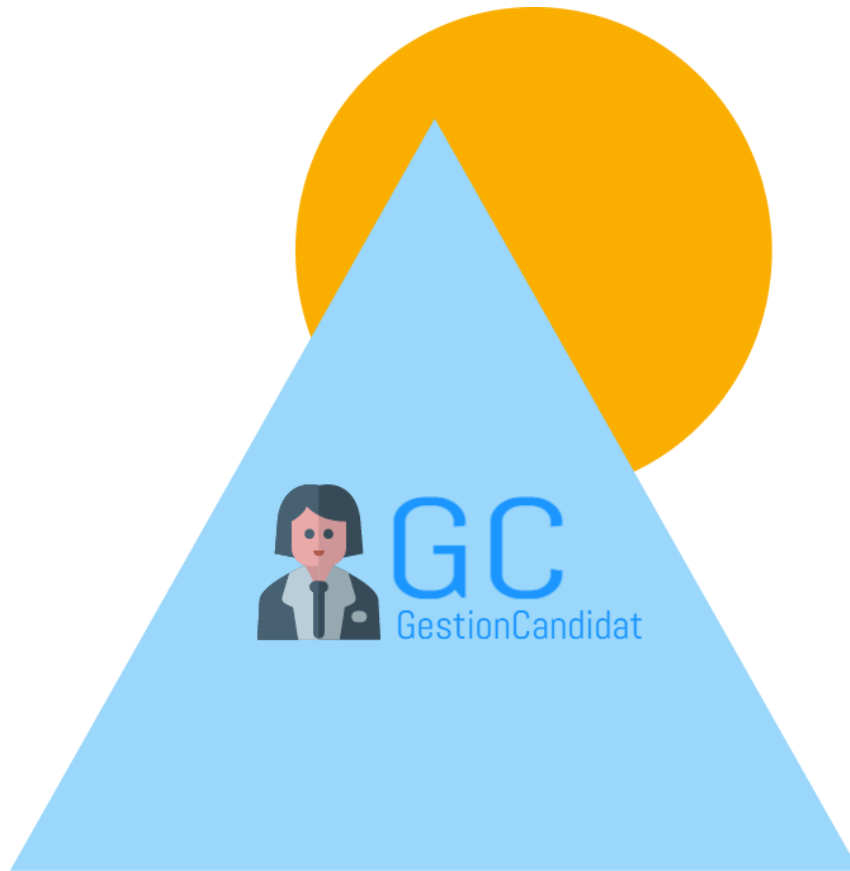


Candidate Management application Android



Sommaire

Introduction

- Contexte
- Objectif

1- Storyboard et Fonctionnalités

- Gestion de candidats
 - Recherche
 - Modification
 - Localisation
 - Envoi de SMS
 - Ajout
- Fonctionnalités de prédiction
 - Prédire la satisfaction
 - Prédire la tranche de salaire
 - Prédire le temps restant dans l'entreprise

2- Documentation technique

- Outils utilisés
- Maintenance de l'application

Introduction

Cette application Androïd a été développée dans le cadre du projet annuel de fin d'année de master à l'ESGI (Ecole Supérieure de Génie Informatique), école du Réseau GES.

Le client final de l'application est la responsable des ressources humaines de l'entreprise C17 Engineering est liée à un serveur développé en .NET Core et déployé sur Azure et est en complément de l'application web et d'une application IOS.

Ce projet répond aux demandes de Julie Goutheraud (RH de chez C17), dans le but de l'aider dans la gestion des candidats.

Dans cette documentation, seront présentées les fonctionnalités techniques ainsi qu'un guide utilisateur afin de faciliter la compréhension et l'utilisation de l'application et enfin une partie technique pour les développeurs, expliquant les outils utilisés et comment maintenir l'application.

Guide utilisateur

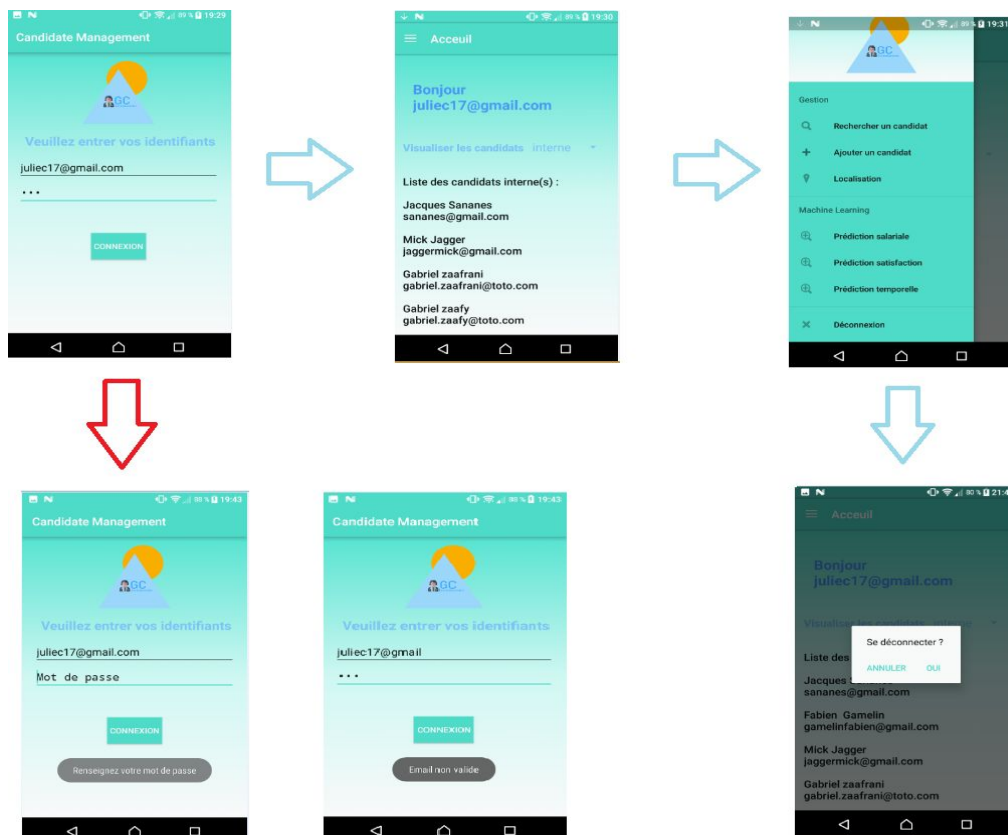
1- Storyboard et Fonctionnalités

Dans un premier temps, sera présentée la navigation pour les fonctionnalités de gestion, puis ensuite les fonctionnalités de prédictions.

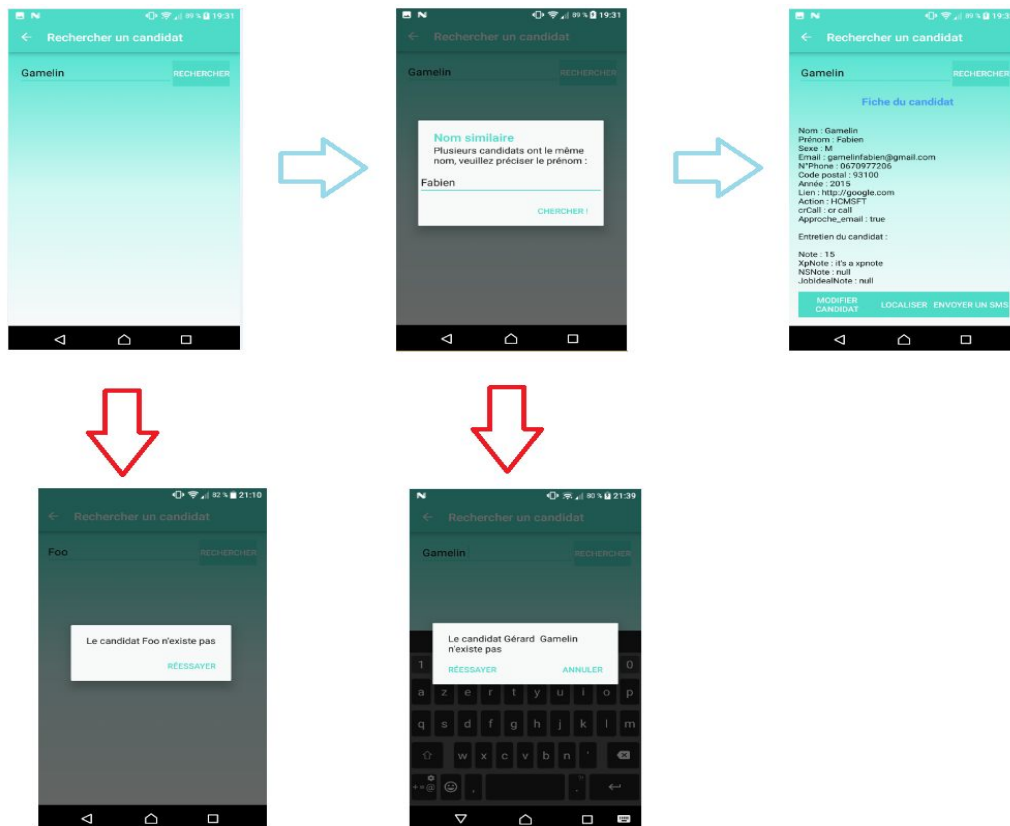
La gestion de candidat n'est pas aussi complète que celle possible dans la version web de l'application, en effet il n'est pas possible d'envoyer, de modifier des reminds par exemple ou d'accéder à l'agenda.

C'est donc une version pocket, plus light permettant de gérer les candidats à distance, de les contacter, localiser etc... afin d'avoir accès aux données partout lorsqu'on souhaite travailler à distance.

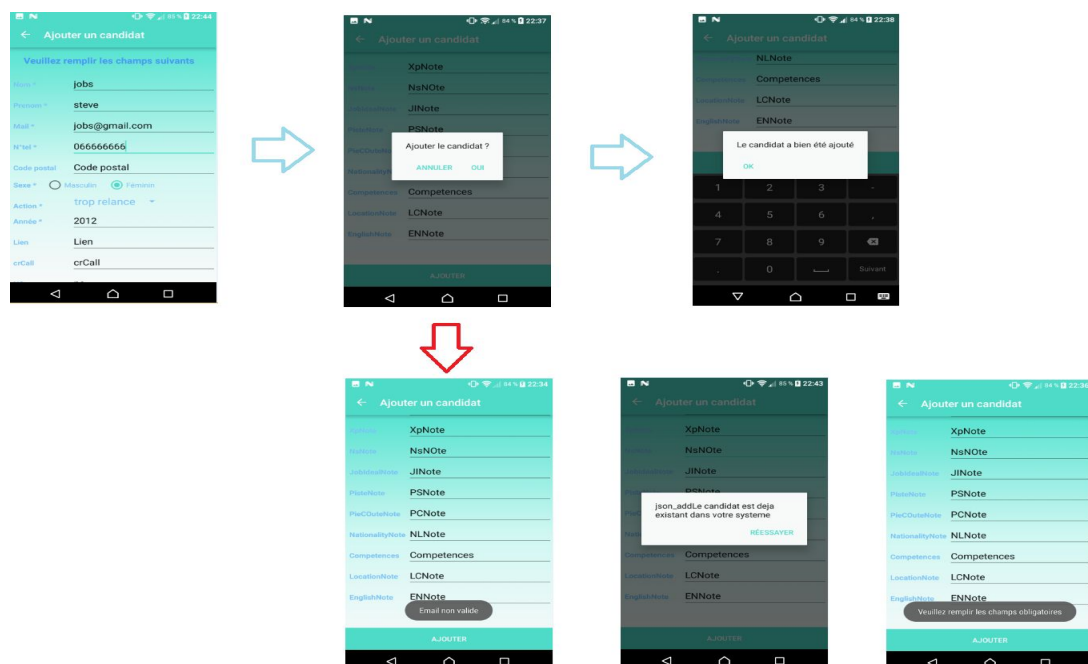
- La connexion
- La recherche par action
 - Permet de rechercher tous les candidats en fonction d'une action choisie



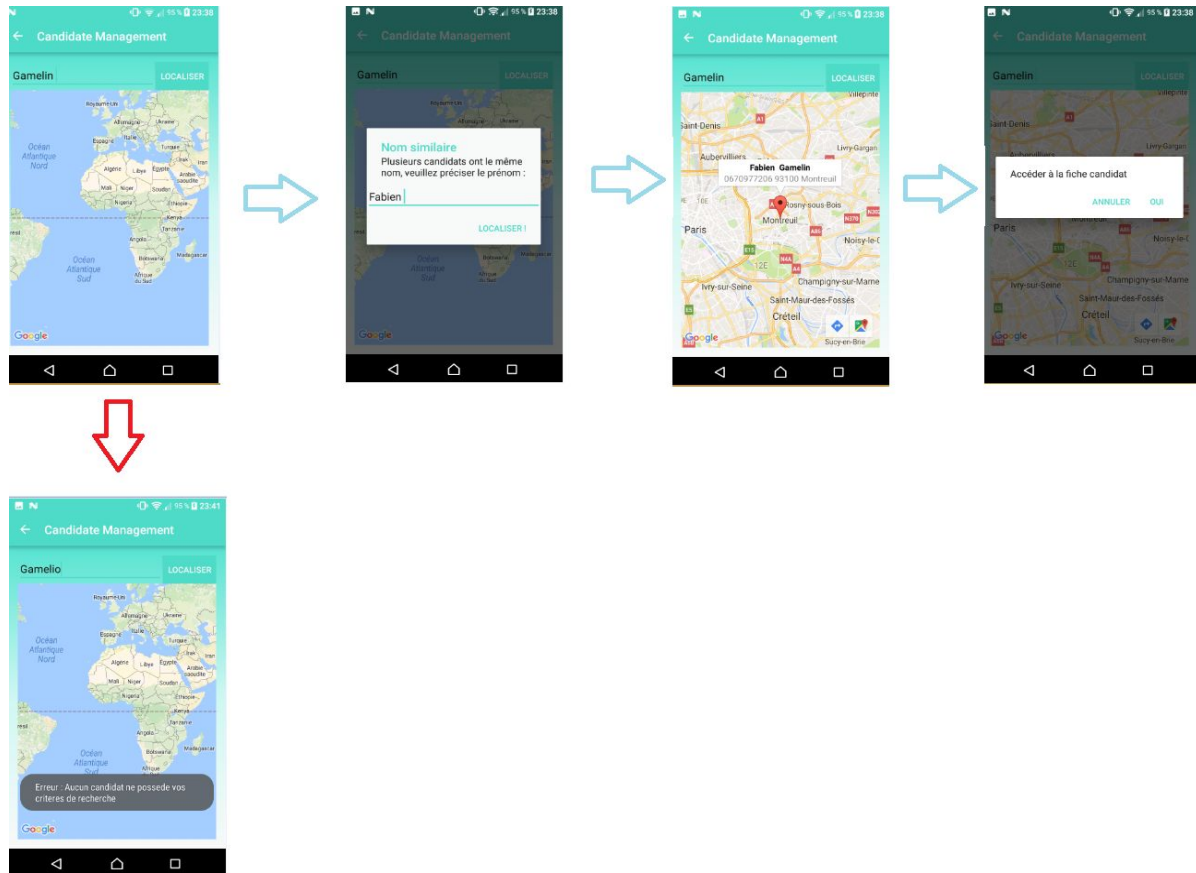
- La recherche individuelle :
 - Permet de rechercher un candidat en fonction de son nom, si plusieurs candidats ont le même nom, il est possible de préciser le prénom.



- L'ajout



- La modification
- La localisation

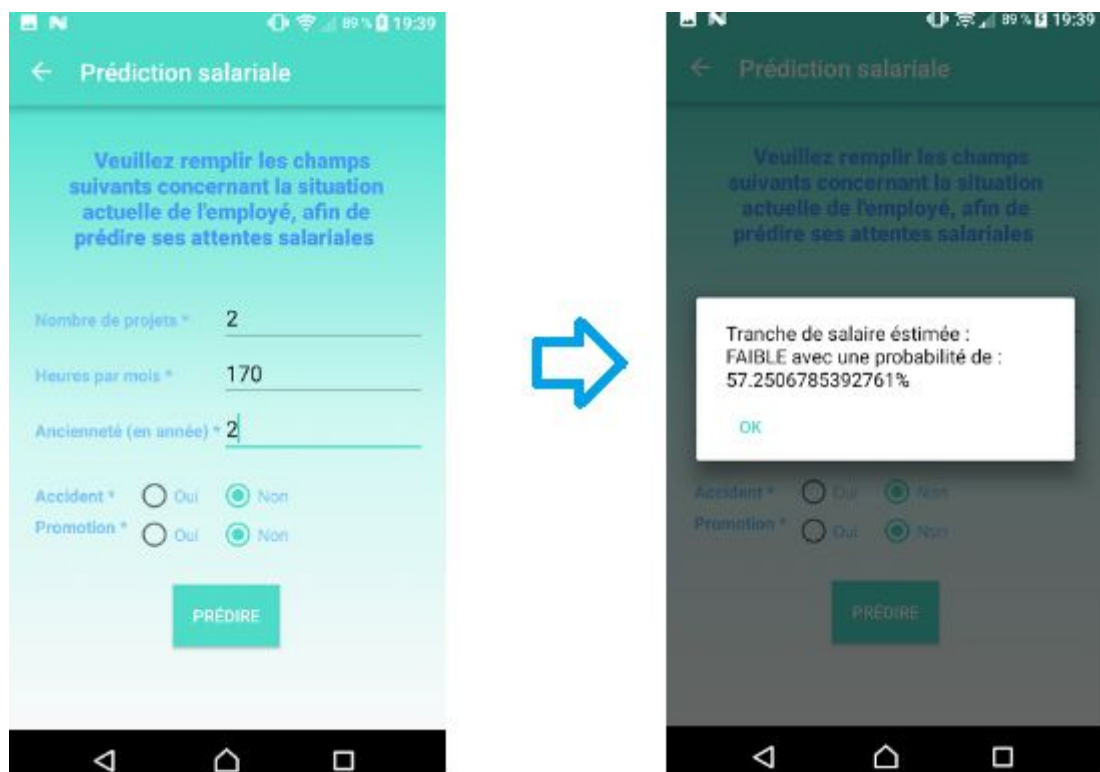


- L'envoi de SMS



La partie prédiction nous offre des fonctionnalités intéressantes (optionnelles du besoin client), en effet grâce à des algorithmes de machine learning ainsi qu'un jeu de données important de candidats existants (Voir Guide Technique plus bas), nous sommes capables de prédire certaines futures actions des employés actuels en fonction de leur situation, voici ces fonctionnalités :

- Prédiction de la satisfaction
 - Permet de prédire la satisfaction d'un employé en fonction de son nombre de projets attribués, son nombre d'heures moyen par mois, son ancienneté, si il a eu un accident durant sa carrière, s'il a déjà obtenu une promotion et sa tranche de salaire.
- Prédiction du nombre d'années dans l'entreprise
 - Permet de prédire le nombre d'années qu'un employé restera (Satisfaction, nombre de projets, nombre d'heures, accident, promotion et salaire)
- Prédiction de la tranche de salaire (pour un employé externe) :
 - Permet de prédire quelle est la tranche de salaire la plus adaptée pour un candidat en fonction de sa situation actuelle (Nombre de projets, nombre d'heures moyen, ancienneté, accident, promotion)



Guide technique

Outils utilisés

L'application a été développée avec Android Studio en java, la version minimum des API android est la 15 (Ice Cream Sandwich) et compilable pour la version 23 (Marshmallow), les librairies utilisées ont été :

- Le framework android (support/widget/util/app/net/content/view/os/text)
- Le framework java api (inclus dans la plateforme android)
- Les librairies de parsing json org
- La librairie Volley pour les requêtes http avec les différents serveurs
- google play services pour la localisation

Maintenance

L'application est déployée sur le Play Store d'Android sous le nom de Candidate Management, le play store fournit donc des rapports réguliers et surtout des crash reports au cas où l'application crasherait à un moment donné, c'est donc très pratique pour le maintien de l'application.

En cas de changement d'adresse du serveur, vous pouvez changer l'adresse comme ceci Dans le fichier APIConstants.java :

```
package com.example.fabiengamel.candidatemanagement.Utils;

/**
 * Created by Fabien gamel on 15/06/2017.
 */
public class APIConstants {

    public static final String BASE_URL = "http://10.33.3.32:5000";
}
```

Modifiez la variable BASE_URL par la nouvelle adresse.

De même, si vous souhaitez modifier/adapter les remind en fonction de ceux en base, vous pouvez modifier le fichier Remind.java

```
public class Remind {

    public static final String PAERemind =
        "Je prends bonnes notes du fait que dans l'immédiat un processus de recrutement au sein de C17 Engineering ne s
        "N'hésitez toutefois pas à m'envoyer votre CV, si vous en avez un à disposition ou lorsque vous le mettrez à jo
        "Je vous propose de vous recontacter dans 6 mois pour faire un point et de me tenir à votre disposition d'ici
        "Je vous souhaite une très bonne continuation d'ici là et vous laisse le loisir de transmettre mes coordonnées
        "Notre fourchette salariale pour une embauche en CDI d'un consultant varie environ entre 35-60 K€ selon le nive
        "Bien à vous et à bientôt ! \n" +
        "Julie ";

    public static final String aRelancerLKD =
        "Merci pour l'ajout sur LinkedIn et enchantée ! Je suis chasseur de tête spécialisée dans le digital, comme vou
        "J'aimerais beaucoup faire un point avec vous à propos de votre situation professionnelle, que vous soyez à l'é
        "Nous pouvons discuter durant 10 minutes par téléphone. A moins que vous ne préfériez discuter par mail ?\n" +
        "Qu'en diriez-vous ?\n" +
        "A bientôt !\n" +
        "Julie";

    public static final String appellerRemind = "vide pour le moment";
    public static final String confirmationRDV =
        "J'ai le plaisir de vous confirmer notre entretien le [Jour 00 Mois] à [00 h00].\n" +
        "Instructions : Interphone C17 Engineering, 7eme étage, code ascenseur : 65412\n" +
        "\n" +
        "Bien à vous,\n" +
        "\n" +
        "Julie.";

}
```

Et modifiez la valeur du remind souhaité

Les prédictions :

Les fonctionnalités de machine learning ont été développés grâce au Saas fournit par le cloud Azure : Azure Machine Learning Studio.

Tout d'abord les données viennent d'un jeu de données candidats de plus de 15000 lignes comprenant plusieurs informations, qui a permit grâce à des algorithmes puissants de machine learning d'entraîner le modèle de données afin de faire des prédictions sur des champs définis en fonction des champs d'entrées choisis.

Exemple : Prédiction temporelle pour un candidat

L'algorithme choisi a été **Multi-Class Forest Decision**.

C'est un algorithme de **classification** multi classes c'est à dire permettant de faire de la **prédiction**, et pas seulement binaire mais plusieurs valeurs

Cet algorithme est réputé comme **précis et rapide**, après avoir en essayé plusieurs, c'est avec celui ci que j'ai le plus de précision (voir "average accuracy" dans le résumé en dessous).

Le fonctionnement se base à partir d'un jeu de données dont il connaît les valeurs, afin de créer un modèle qui prédit la valeur d'une variable-cible depuis la valeur de plusieurs variables d'entrées.

Une des variables d'entrée est sélectionnée à chaque nœud interne (nœud qui n'est pas terminal) de l'arbre selon une méthode qui dépend de l'algorithme.

Chaque arête vers un nœud-fils correspond à un ensemble de valeurs d'une variable d'entrée, de manière à ce que l'ensemble des arêtes vers les nœuds-fils couvrent toutes les valeurs possibles de la variable d'entrée.

Cependant c'est un algorithme lourd, puisqu'on cherche à chaque nœud de l'arbre le partage optimal, dans le but d'obtenir le meilleur partage possible sur l'ensemble de l'arbre de décision. C'est la stratégie la plus commune pour apprendre les arbres de décision depuis les données.



Création du modèle entraîné et transformation en web service

Metrics

Overall accuracy	0.522603
Average accuracy	0.880651
Micro-averaged precision	0.522603
Macro-averaged precision	0.490886
Micro-averaged recall	0.522603
Macro-averaged recall	0.451833

Confusion Matrix

		Predicted Class							
		2	3	4	5	6	7	8	10
Actual Class	2	38.0%	46.0%	9.5%	2.5%	2.0%	0.6%	0.9%	0.5%
	3	23.2%	64.5%	7.0%	2.1%	1.4%	0.6%	0.5%	0.7%
	4	17.5%	26.2%	46.1%	6.1%	2.8%	0.5%	0.3%	0.5%
	5	9.1%	14.1%	12.0%	55.5%	8.6%	0.5%	0.1%	
	6	14.2%	20.8%	9.4%	26.2%	27.1%	1.1%	0.9%	0.3%
	7	23.7%	18.6%	3.1%	2.1%	4.1%	46.4%		2.1%
	8	19.0%	26.6%	3.8%		5.1%	2.5%	43.0%	
	10	11.3%	32.2%	8.7%	5.2%	1.7%			40.9%

Résumé des probabilités par valeur

Une fois ces modèles fonctionnels et répondants aux attentes de prédiction, azure donne la possibilité de les déployer en tant que web services.

Les webservices sont alors appelés dans l'application avec les données d'entrées adaptées et l'autorisation nécessaire.

Request

Method	Request URI	HTTP Version
POST	https://ussouthcentral.services.azureml.net/workspaces/5587f0c68bac40fcb8ad1bf09cdc2c99/services/8984cc55e3fa40c0b07601ff8b98969f/exe cute?api-version=2.0&details=true	HTTP/1.1

Note: You may omit the **details** parameter from the query string. This would cause **ColumnTypes** to be omitted from the output

Request Headers

Request Header	Description
Authorization:Bearer abc123	Required. Pass the API Key here. Obtain this key from the publisher of the API.
Content-Length	Required. The length of the content body.
Content-Type:application/json	Required if the request body is sent in JSON format.
Accept: application/json	Optional. Use the header to receive the response in JSON format.

Url et headers de la requête

Request Body

Sample Request

```
{
  "Inputs": {
    "input1": {
      "ColumnNames": [
        "satisfaction_level",
        "number_project",
        "average_monthly_hours",
        "Work_accident",
        "left",
        "promotion_last_5years",
        "salary"
      ],
      "Values": [
        [
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "value"
        ],
        [
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "value"
        ]
      ]
    }
  },
  "GlobalParameters": {}
}
```

Body (raw json) de la request

input1

number_project

3

average_monthly_hours

200

time_spend_company

3

Work_accident

0

promotion_last_5years

0

salary

low

Test Request-Response

output1

Scored Labels 0.55

Test de la requête