



RAPPORT DE PROJET

Classement d'hôtels pour un
voyage



MICHAEL CHEN & ALEXANDRE YOU

Table des matières

1	<i>Définition du problème et de nos ressources</i>	2
1.1	<i>Motivation</i>	2
1.2	<i>Ressources</i>	2
2	<i>Notre démarche</i>	4
2.1	<i>Les actions potentielles</i>	4
2.2	<i>Définition de notre famille de critère</i>	4
3	<i>Les stratégies adoptés</i>	6
3.1	<i>Somme pondérée</i>	6
3.2	<i>Electre IS</i>	6
4	<i>Implémentation</i>	8
4.1	<i>Somme pondérée</i>	8
4.2	<i>ELECTRE IS</i>	8
5	<i>Résultats</i>	9
5.1	<i>Interprétation</i>	9
5.2	<i>Critique</i>	9

1 *Définition du problème et de nos ressources*

1.1 *Motivation*

Comme des millions de Français et de voyageurs à travers le monde, partir en voyage est une source de plaisir et d'évasion, mais trouver le bon hôtel peut vite devenir un casse-tête. Face à la multitude de choix disponibles, la réservation d'hébergement est souvent une étape chronophage et stressante, surtout lorsqu'il s'agit de faire le meilleur choix en fonction de critères personnels.

Étudier cette problématique, c'est répondre à un besoin concret : aider chaque voyageur à profiter pleinement de son séjour en lui simplifiant l'accès à des options d'hébergement qui correspondent à ses attentes. Dans un monde où le temps est précieux et où l'expérience utilisateur est au cœur des priorités, une solution qui facilite la recherche d'hôtel en fonction des préférences individuelles peut faire toute la différence.

C'est donc l'objectif de notre projet que d'essayer de répertorier les solutions optimales adaptées à n'importe quel profil. Pour cela nous allons voir 2 méthodes différentes pour classer les actions avec leurs avantages et inconvénients. Ces méthodes diffèrent des classements classiques qui dépendent uniquement de filtres et généralement d'une seule caractéristique.

1.2 *Ressources*

Avant de présenter le dataset que nous avons choisi, nous voulions vous faire part d'une remarque : Les hôtels se décrivent eux-mêmes dans leur annonce et manque parfois d'objectivité. Par souci de justesse, nous avons donc jugé préférable, et plus sûr, de considérer une base de données contenant des informations venant d'avis externes et d'anciens clients plutôt qu'une autre avec des descriptions d'hôtels potentiellement romancées.

Notre dataset nous vient du site Kaggle, partagé par **Jiashen Liu** et intitulé *515K Hotel Reviews Data in Europe*. Cette base de données contient les avis de 515 000 clients pour 1493 hôtels européens distincts sur un total de 17 dimensions :

- **Hotel_Address** : Adresse de l'hôtel. Il s'agit d'une adresse complète contenant aussi l'arrondissement, la ville et le pays.
- **Additional_Number_of_Scoring** : Nombre supplémentaire d'avis valides sans commentaire.
- **Review_Date** : Date à laquelle l'évaluateur a publié l'évaluation.
- **Average_Score** : Note moyenne de l'hôtel.
- **Hotel_Name** : Nom de l'hôtel
- **Reviewer_Nationality** : Nationalité de l'évaluateur.
- **Negative_Review** : Avis négatif que l'auteur de l'avis a donné à l'hôtel. Si l'auteur de l'évaluation ne donne pas d'évaluation négative, il sera indiqué « No Negative »
- **Review_Total_Negative_Word_Counts** : Nombre total de mots dans l'évaluation négative.
- **Positive_Review** : Avis positif que l'auteur de l'avis a donné à l'hôtel. Si l'auteur de l'évaluation ne donne pas d'évaluation positive, il sera indiqué : 'No Positive'

- `Review_Total_Positive_Word_Counts` : Nombre total de mots dans l'évaluation positive.
- `Reviewer_Score` : Note attribuée à l'hôtel par l'auteur de l'évaluation, sur la base de son expérience.
- `Total_Number_of_Reviews_Reviewer_Has_Given` : Nombre d'évaluations que l'auteur de l'évaluation a données dans le passé.
- `Total_Number_of_Reviews` : Nombre total d'avis valides sur l'hôtel.
- `Tags` : Tags que l'auteur de l'évaluation a attribué à l'hôtel.
- `days_since_review` : Durée écoulée en jours depuis la date de l'évaluation
- `lat` : Latitude de l'hôtel
- `lng` : Longitude de l'hôtel

Dans la suite, nous n'allons sélectionner et classer seulement les hôtels qui se situent dans la ville de destination du voyageur. Notre objectif sera de classer ces hôtels du meilleur au pire pour guider le décideur dans sa prise de décision

2 Notre démarche

2.1 Les actions potentielles

Dans notre problème de choix d'hôtels, il semble évident que notre liste d'actions potentielles s'étend au choix d'un hôtel parmi tout ceux disponible dans la ville du séjour. Choisir plus d'un hôtel desservirai notre décideur, tout comme ne pas en choisir du tout.

2.2 Définition de notre famille de critère

Pour ce qui est de notre famille de critère, nous avons fait preuve de créativité. Remarquez d'abord que notre objectif est de construire un classement des meilleurs hôtels, il n'est donc pas intéressant de prendre en compte le critère **Average_Score** qui donne directement une note à l'hôtel, nous allons ignorer ce critère. Nous voulions tout de même utiliser ce critère dans un but de comparaison avec nos résultats mais la note ne peut prendre qu'un nombre trop limité de valeur si bien qu'il n'est pas rare de voir un grand nombre d'hôtel partagé la même note. D'autres sont aussi complètement non-significatif (la nationalité de l'auteur du commentaire par exemple).

Enfin il y a ceux qui ont besoin d'une transformation, par exemple le critère **Positive_Review** comporte des informations utiles mais pas directement applicables. On appliquera le feature engineering pour expliciter les caractéristiques appropriés à notre choix d'hôtel.

Dans un tout premier temps, spécifions la requête du décideur. Elle doit expliciter les caractéristiques voulues par le décideur ainsi qu'une ville. Un exemple : "Je veux un hôtel très luxueux avec une vue sur la Tour Eiffel à Paris". A l'aide du modèle NLP, on extrait des informations à partir de cette entrée utilisateur en langage naturel et stocke la ville et les différentes caractéristiques dans un dictionnaire. Ensuite, pour ces mots clés, tous obtiennent un score de base et une pondération proportionnelle qui vont être directement utilisés dans les algorithmes de décision multicritère

Présentons, sans plus attendre, le premier critère évalué dans notre étude, la pénalité. On calcule pour chaque hôtel, sa pénalité. Pour cela, on passe en revue tous les tags et les commentaires existants et pour chaque caractéristique présent dans la requête du décideur, si elle n'apparaît jamais ni dans les tags ni les avis, on incrémente cette pénalité de 1. On multiplie le score final par -0.2 car il s'agit d'une pénalité donc négatif et pour réduire l'effet de celle-ci.

Les deux critères suivants sont l'impact positif et négatif. On définit 5 mots clés avec chacun un poids de 0.2 et on recherche ces mots dans tous les commentaires d'un même hôtel. Ces "critères universels" pour l'impact positif sont ["staff", "clean", "breakfast", "wifi", "quiet"]. Ainsi dès qu'un des 5 mots se retrouvent dans un commentaires positif de l'hôtel, on rajoute 0.2 au score d'impact positif pour cette hôtel. Le même procédé est fait pour l'impact négatif avec la liste suivante à retrouver dans les commentaires négatifs : ["wifi", "dirty", "staff", "noisy", "breakfast"]. Si ces mots se retrouvent dans les commentaires négatifs, ils ont une mauvaise connotation. On ajoute 0.2 à chaque occurrence dans le score d'impact négatif.

Ensuite viens le score dynamique. Contrairement aux deux critères précédents qui utilise des critères "universels", celui-ci se base sur les caractéristiques demandés par le décideur dans sa requête. Similairement, dès qu'une caractéristique se retrouve dans un avis positif, le score dynamique s'en retrouve incrémenté du score associé à la dite caractéristique. On retire ce score si la caractéristique est retrouvée dans un commentaire négatif. Si une caractéristique fait parti de ses tags, on ajoute 2 fois le score au score dynamique.

Par exemple, « Je veux trouver un hôtel de luxe à Paris qui convienne aux familles ». Nous obtenons alors deux exigences : luxe et convivialité, avec une pondération de 0,5 pour chacune d'entre elles. Ensuite, on associe les mots-clés qui correspondent aux exigences. C'est-à-dire « luxe », « haut de gamme », « premium... » pour l'exigence de luxe et « famille », « enfants », « convivialité pour les enfants », « aire de jeux... » pour l'exigence de famille. Chaque fois qu'il y a une correspondance du mot-clé dans le commentaire, nous ajoutons 1 point à ce commentaire. S'il y a une correspondance dans l'étiquette, nous ajoutons 2 points au commentaire. Après avoir passé en revue tous les commentaires, nous les regroupons par nom d'hôtel et calculons la somme (la raison pour laquelle nous n'utilisons pas la moyenne est qu'il y a beaucoup d'avis qui totalisent 0 point). Enfin, nous normalisons le score dynamique de tous les hôtels.

Le prochain critère est le nombre d'avis d'un hôtel. On compte tout simplement le nombre d'avis total pour un hôtel donné, en appliquant tout de fois une transformation logarithmique pour donné plus d'importance aux premiers avis. Une indifférence sur le critère de nombre d'avis est considéré par exemple lorsqu'on compare un hôtel à 15000 avis et un autre à 14500.

On peut remarquer que la plupart des critères sont construit à l'aide d'une somme pondérée. Voici donc notre famille de critères.

3 *Les stratégies adoptées*

Pour l'exploitation des données et la construction du classement, nous avons utilisé deux méthodes différentes amenant chacun à un classement différent. Ces deux méthodes ont leurs avantages et leurs inconvénients. Donnons les principes de ces deux méthodes.

3.1 *Somme pondérée*

La première méthode utilisée est la somme pondérée. Elle consiste à faire la somme des performances de chaque hôtel, pondérées par un jeu de poids. Attention tout de même, une phase de transformation des données est parfois nécessaire. En effet, des critères peuvent ne pas être noté sur des échelles similaires. Certains prenant des valeurs astronomiques et d'autres sont seulement des variables binaires. Ainsi la note finale s'en retrouve affecté, une note pouvant prendre de "gros valeurs" prendra plus d'importance qu'une autre ne prenant ses valeurs que dans faibles quantités, toute chose étant égale par ailleurs. Pour régler cela, la normalisation pourrait être une solution viable.

Cette méthode a pour avantage d'être très facile à implémenter et son exécution est rapide. En effet avec la somme pondérée, chaque hôtel se voit attribuer une note unique qu'il suffira de considérer à la place de toutes ses performances sur les différents critères. Ainsi pour faire notre classement, il suffira de sélectionner les meilleurs hôtels en fonction de la somme pondérée obtenue.

Cependant, la somme pondérée présente plusieurs inconvénients non négligeables :

En premier lieu, la compensation. La somme pondérée permet l'équilibrage de mauvaises performances dans un critère par de bonnes dans un autre. Elle ignore aussi les interdépendances. La somme pondérée suppose que les critères sont indépendants les uns des autres, ce qui n'est pas toujours le cas dans des systèmes complexes. Il y aussi des risques de surpondération ou sous-pondération due à une détermination des poids subjective et qui ne reflète pas l'importance réelle des critères.

3.2 *Electre IS*

La deuxième méthode utilisée est la relation de surclassement Electre IS. Commençons par expliquer Electre IS. C'est une approche utilisée, tout comme la somme pondérée, pour comparer et classer plusieurs alternatives en tenant compte de plusieurs critères. La méthode repose sur la notion de surclassement, qui exprime dans quelle mesure une alternative A peut être considérée comme "au moins aussi bonne" qu'une autre alternative B.

La relation de surclassement dans ELECTRE IS est établie à partir de deux concepts fondamentaux : le concordance et la discordance.

Le critère de concordance évalue si une alternative A est au moins aussi bonne que B, selon un critère donné. Chaque critère est associé à un poids, qui reflète son importance relative. On calcule alors l'indice de concordance global $C(A,B)$, qui mesure la proportion

des critères en faveur du surclassement de A sur B. Si elle dépasse un seuil de concordance donné, alors on peut évaluer le critère de discordance pour ce couple de profil.

Le critère de discordance identifie si un ou plusieurs critères s'opposent fortement au surclassement de A sur B. A l'aide d'un seuil de veto, on mesure dans quelle mesure A est moins bon que B sur ce critère. Si la discordance pour un critère dépasse ce seuil de veto, le surclassement de A sur B est contesté.

Ainsi, A surclasse B si $C(A,B)$ est supérieur au seuil de concordance et qu'aucun critère n'oppose son veto.

On peut ajouter deux autres seuils. Le premier est le seuil d'indifférence qui représente la différence maximale acceptable entre deux alternatives pour qu'elles soient considérées comme équivalentes du point de vue du décideur. Le deuxième est le seuil de préférence au-delà duquel une alternative est nettement préférée par rapport à une autre sur un critère donné.

Electre permet donc une grande liberté au décideur en lui laissant le choix des seuils pour chaque critère. Elle réduit aussi considérablement l'effet compensatoire en appliquant le critère de non-discordance.

Cependant, Electre IS a aussi ses défauts. Tout d'abord, le temps d'exécution est affecté ce qui est naturel puisque le nombre de calculs à réaliser est bien plus important qu'avec une somme pondérée. Aussi, trouver des bons seuils est un art. Si les seuils sont trop larges, cela ne nous permettra pas d'établir assez de domination de surclassement. S'ils sont trop stricts, ils peuvent rendre le processus de décision trop rigide.

4 *Implémentation*

4.1 *Somme pondérée*

Le calcul des scores des hôtels avec une somme pondérée est très simple. On considère une telle hiérarchie des critères :

$$\text{Dynamic_Score} > \text{Penalty} > \text{Positive_Impact} = \text{Negative_Impact} > \text{Scaled_Reviews}$$

On donne des poids aux critères qui respectent la hiérarchie et notre score finale est donc donné par :

$$0.45 \times \text{Dynamic_Score} + 0.3 \times \text{Penalty} + 0.1 \times \text{Positive_Impact} + 0.1 \times \text{Negative_Impact} + 0.05 \times \text{Scaled_Reviews}$$

4.2 *ELECTRE IS*

Comme nous l'avons expliqué, pour cette méthode nous devons vérifier les conditions de concordance et de non-discordance. Pour cela, on va créer les matrices suivantes : une matrice de concordance, une matrice de discordance et enfin une matrice de surclassement.

Commençons par la matrice de concordance. Elle mesure, pour chaque paire d'hôtels i et j , le degré auquel i est au moins aussi bon que j sur tous les critères. Pour toute paire d'hôtel i et j , on compare chaque critère pour i et j , on compte le pourcentage de critères où i est meilleur ou égal à j et on place ce pourcentage dans la matrice à la i ème ligne et j ème colonne.

Passons ensuite à la matrice de discordance. Elle mesure, pour chaque paire d'hôtels i et j , l'ampleur maximale d'une différence où j surpasse i . Ainsi, pour chaque paire, on calcule pour chaque critère la différence entre j et i , on identifie la plus grande différence puisqu'il suffit qu'un critère pose son veto pour que le surclassement est rejeté et on place la valeur de cette plus grande différence dans la matrice à la i ème ligne et j ème colonne.

Enfin la matrice de domination qui combine les informations de concordance et de discordance pour indiquer si i domine j . On introduit un seuil de concordance (ici 0.6) et un seuil de veto (ici 0.3). Pour chaque paire, i domine j si la valeur associée dans la matrice de concordance est supérieur ou égale au seuil de concordance ET la valeur associée dans la matrice de discordance est inférieur ou égale au seuil de veto. Si les deux conditions sont validés, on place la valeur booléenne TRUE dans la matrice de domination, FALSE sinon.

Un préordre complet se dessine. On calcule pour chaque hôtel, le nombre d'hôtel qu'il surclasse, plus ce nombre est élevé, plus l'hôtel en question est haut dans le classement. On a ainsi un nouveau classement.

```
user_input = "I want a luxury family hotel in Paris near the city center."
```

Résultats avec la somme pondérée

Hotel_Name	Final_Score
H tel Regina	0.494849
Saint James Albany Paris Hotel Spa	0.376025
H tel Aiglon Esprit de France	0.324972
H tel Bedford	0.285490
Novotel Paris Les Halles	0.278172
Residence Henri IV	0.262852
H tel D Aubusson	0.259208
Maison Albar Hotel Paris Op ra Diamond	0.247058
Hotel d Orsay Esprit de France	0.241687
Royal Saint Michel	0.227168

Résultats avec Electre IS

```
Hotels Ranked by Dominance:  
1. H tel des Academies et des Arts - Dominance Score: 103  
2. H tel Recamier - Dominance Score: 103  
3. Le G n ral H tel - Dominance Score: 102  
4. Madeleine Plaza - Dominance Score: 91  
5. Le Relais Madeleine - Dominance Score: 90  
6. Sofitel Paris Le Faubourg - Dominance Score: 89  
7. Splendide Royal Paris - Dominance Score: 88  
8. L H tel - Dominance Score: 86  
9. Hotel Plaza Athenee Paris - Dominance Score: 84  
10. H tel Regina - Dominance Score: 81
```

```
user_input = "I want a romantic hotel in Paris with rich nightlife and good view."
```

Résultats avec la somme pondérée

Hotel_Name	Final_Score
Pullman Paris Tour Eiffel	0.377109
Holiday Inn Paris Notre Dame	0.290621
Terrass H tel Montmartre by MH	0.240595
Villa Beaumarchais	0.225502
Hyatt Regency Paris Etoile	0.206315
Le Parisis Paris Tour Eiffel	0.200910
Little Palace Hotel	0.175314
H tel Regina	0.172269
Mercure Paris Terminus Nord	0.164845
Hotel Mademoiselle	0.143779

Résultats avec Electre IS

```
Hotels Ranked by Dominance:
1. Artus Hotel by MH - Dominance Score: 72
2. Hotel Square - Dominance Score: 69
3. Hotel La Villa Saint Germain Des Pr s - Dominance Score: 61
4. Nell Hotel Suites - Dominance Score: 55
5. Best Western Plus 61 Paris Nation Hotel - Dominance Score: 54
6. Le Metropolitan a Tribute Portfolio Hotel - Dominance Score: 53
7. Goralska R sidences H tel Paris Bastille - Dominance Score: 51
8. B Montmartre - Dominance Score: 48
9. Mercure Paris Porte De Versailles Expo - Dominance Score: 47
10. Hotel Dupond Smith - Dominance Score: 46
```

5 *Résultats*

5.1 *Interprétation*

Les résultats pour ces deux méthodes donne des résultats complètement différents. En effet, dans les 10 premiers hôtels fournis, seul l'hôtel Regina apparaît dans les listes et seulement pour la première requête. Alors pourquoi des résultats si différents ? En plus de procéder de manières différentes, les deux méthodes ne prennent pas en compte les mêmes critères. L'un prend le score dynamique, l'impact positif, l'impact négatif, la pénalité et le nombre de commentaires alors que pour l'autre on ignore ces critères et on ne prendra uniquement les caractéristiques dans la requête du décideur, et il faudra que les avis de l'hôtel recense ces caractéristiques le plus possible. De plus, Electre IS dispose de différents seuils qui ne sont pas pris en compte dans le calcul dans la somme pondérée.

Pour être sur, on a calculé la corrélation de Spearman pour comparer les deux méthodes de classement et voir si elles produisent des classements similaires (50 premiers). Comment interpréter cette corrélation ? Elle varie entre -1 et 1. Si elle est égale à 1, la corrélation entre les deux classements est parfaite, si elle est égale à 0 il n'y a aucune corrélation. Pour les deux requêtes, on a respectivement les corrélations de Spearman suivantes : 0.5167 et 0.1152.

Le premier test donne une similarité beaucoup plus élevée que le second. Cela s'explique par le fait qu'Electre crée des scores redondants, ce qui signifie qu'il n'arrive pas à classer deux hôtels qui sont également dominants. Par conséquent, alors que l'Electre IS est plus robuste et peut éviter les scénarios où une exigence de l'utilisateur est trop faible pour l'hôtel, il a des difficultés à classer les hôtels lorsque l'exigence est vague. La somme pondérée résout le problème en ajoutant d'autres critères tels que les critères universels.

Une autre observation est que les hôtels qui ont moins d'avis sont susceptibles d'être moins bien classés, ce qui est logique à la fois dans le modèle et dans la réalité. Étant donné que les modèles additionnent les points lorsqu'il y a des correspondances, les hôtels qui ont plus d'avis sont mieux classés dans la somme pondérée avec la caractéristique du nombre d'avis, et ils sont plus dominants avec des valeurs plus élevées dans le tableau des caractéristiques.

5.2 *Critique*

Commençons par établir que les résultats obtenus dépendent fortement des critères utilisés ainsi que leurs poids. Changer les seuils changerait aussi le classement. On pourrait aussi affirmer sans problème qu'une autre approche aurait donnée une différente prédiction.

Comme évoqué dans la partie précédente, la somme pondérée est bien plus rapide qu'ELECTRE IS en terme de temps d'exécution. Cela fait du sens puisqu'Electre IS est basé sur une analyse paire par paire. Aussi, la somme pondérée fournit une évaluation simple et globale.

Electre IS, bien que plus difficile à interpréter, est beaucoup plus robuste due à son

fonctionnement. En d'autres termes, la somme pondérée est plus simple mais Electre IS est plus robuste.

Les deux méthodes ont donc leurs forces et faiblesses. Le choix dépendra des priorités et de la compréhension du décideur. Une approche serait de combiner les deux méthodes, d'abord utiliser la somme pondérée pour avoir une vue d'ensemble, puis affiner en appliquant Electre IS en cas de priorité forte.