Challenge 2: Apprentissage Supervisé

Ricardo FERNANDEZ | Sarra MAHMOUDI | Chenjie QIAN | Soumiya RAZZOUK | Daniel TERAN

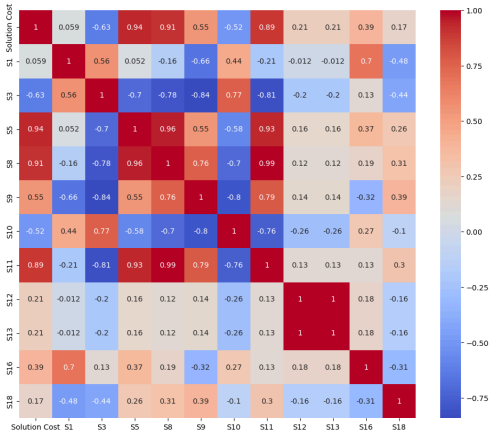
1. Statistiques descriptives et feature engineering

Loading and cleaning

Nous avons commencé par importer nos données, et les rassembler dans une base de données df en ajoutant les colonnes Instance name et Solution cost.
Nous avons constaté que la colonne ‘S7’ n’a pas été calculée, nous l’avons supprimé de la database.

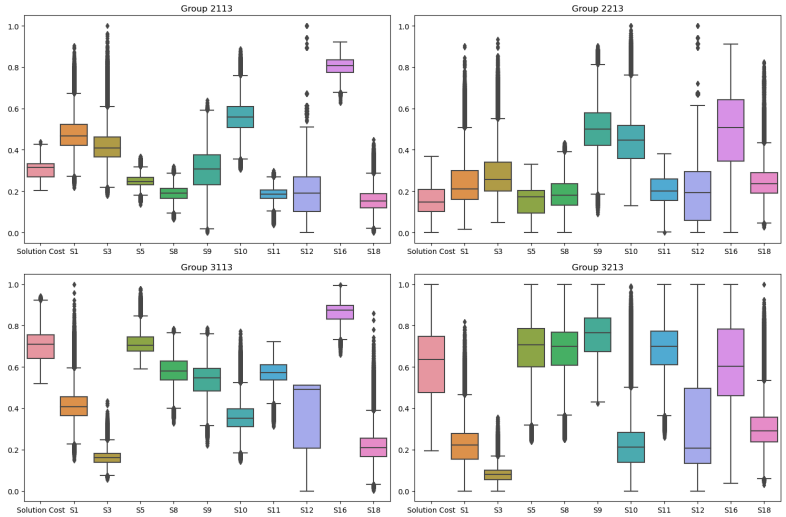
Nous avons vérifié que df ne contient aucune valeur nulle et après nous avons choisi de supprimer toutes les variables statistiques ‘S2’, ‘S4’, ‘S6’, ‘S14’, ‘S15’, ‘S17’, car elles seront fortement corrélées avec les autres variables.

Une analyse des corrélations a montré une grande corrélation entre ‘S12’ et ‘S13’ (coefficient = 1), quitte à supprimer ‘S13’. Et les corrélations suivantes :




```
Correlation with the Target Variable:
Solution Cost    1.000000
S5               0.936667
S8               0.912256
S11              0.894752
S9               0.551346
S16              0.387145
S13              0.213385
S12              0.213364
S18              0.171649
S1               0.058614
S10             -0.516273
S3              -0.631101
```

À ce stade, afin de mieux appréhender les différents comportement des données, nous avons procédé à leur normalisation.



Sampling the data

Le volume de notre data requiert de choisir un échantillon à étudier. Une comparaison statistique entre différents échantillons de 99%, 10%, 5%, 1%, 0.1%, 0.01% montre que même un échantillon de 0.01% sera représentatif de notre data et qu'on va perdre l'information.
Pour cela, **nous avons sélectionné 0.01%** de chaque fichier csv et nous avons combiné les datasets comme avant.

	Sample Percentage	Mean Solution Cost	Std Solution Cost	\
0	0.9900	17237.758200	5029.207493	
1	0.1000	17237.917210	5029.469393	
2	0.0500	17237.991569	5029.733449	
3	0.0100	17238.025848	5028.379146	
4	0.0010	17239.764760	5028.850203	
5	0.0001	17244.472340	5036.059851	
6	0.0010	17238.237366	5029.943409	

	Max Solution Cost	Min Solution Cost
0	27588	7766
1	27356	7769
2	27132	7766
3	26883	7774
4	26599	7780
5	26619	7795
6	26744	7782

2. Benchmark des méthodes de régression pour prédire le coût d'une solution d'un problème CVRP

Pour répondre à la question de prédiction de la variable coût, nous avons utilisé différents modèles de régression, et nous avons procédé par comparaison des performances pour sélectionner le modèle le plus performant en termes de généralisation. Nous avons divisé notre échantillon en 80% pour l'entraînement et 20% pour le test. Et pour tester la capacité de généralisation de nos modèles, nous avons utilisé la cross-validation.

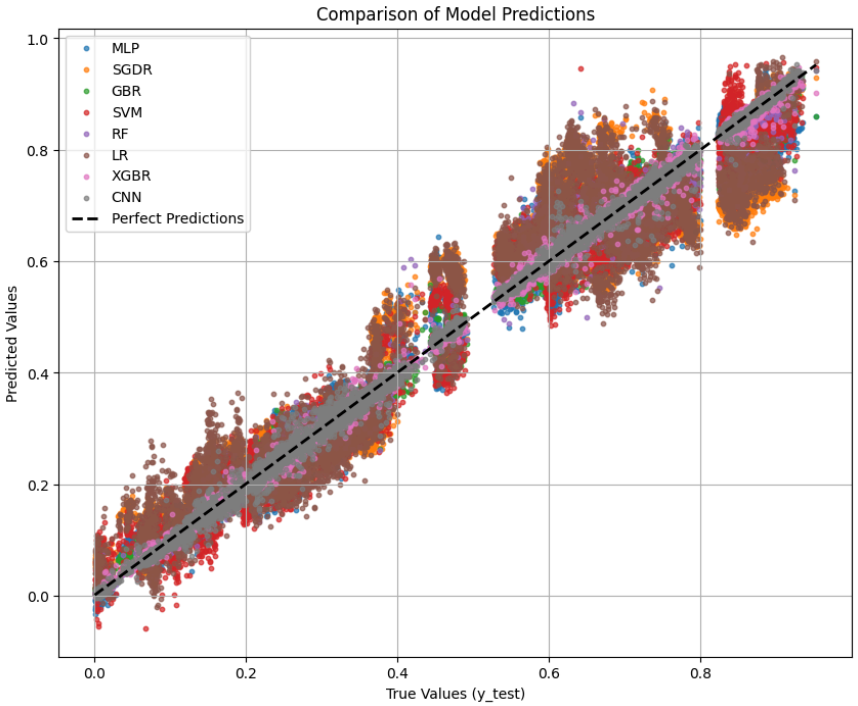
Définition des métriques utilisés :

RMSE (Root Mean Squared Error) - EQM racine carrée , MAE (Mean Absolute Error) - EAM, MSE (Mean Squared Error) - EQM. R-Square score (Coefficient de détermination)

Le tableau ci-dessous résume les résultats de notre étude :

Modèle	Résultat				Cross-validation		
	MAE	MSE	RMSE	R^2	RMSE	MSE	MAE
Linear Regression Model	0,05686	0,00523	0,07230	0,92223	0,07300	X	0,05890
Random forest	X	0,00000	X	1,00000	X	0,02660	0,01840
SVM (différents paramètres)	X	0,00252	X	X	0,06000	X	0,05170
Gradient Boosting Regressor		0,00028			0,02620	X	0,05170
XGBRegressor	X	X	X	X	X	X	X
SGDRegressor		0,00612		0,90434	0,08100	X	0,06570
Multi layer perceptron	0,02739	X	0,03738		0,04420	X	0,03440
Convolutional Neural Network	0,00736	X	0,01072	X	X	X	X

Le graphe ci-dessous, établit une comparaison entre les modèles :



A partir du graphe et du tableau, on remarque que tous les **modèles utilisés font de bonnes prédictions et peuvent généraliser** la connaissance sur de nouvelles données de Test. Et on retient **Convolutional Neural Network, Random Forest et XGB Regressor comme modèles les plus performants (Voir Jupyter Notebook).**

3. Transformation en un problème de classification

Nous réalisons un **échantillon plus grand (25%)** pour explorer la distribution des données et sélectionner la meilleure alternative pour la classification et la répartition des étiquettes. L'échantillon **prend en compte le groupe auquel l'instance appartient** (soit 2113, 2213, 3113 ou 3213). Aucune valeur manquante n'a été trouvée, donc nous procédons à normaliser l'échantillon.

Dans le processus de normalisation, nous décidons d'utiliser un **MinMax normalizer,** en choisissant le meilleur résultat comme 0 et le pire comme 1, et de le répartir à travers chaque groupe, étant donné que les conditions sur chaque regroupement étaient différentes. Ainsi, si nous voulons un modèle généralisable, nous devons standardiser chaque partie de données individuellement avec ses maximums et minimums respectifs, **en considérant que le coût de la meilleure solution est le plus petit.**

Categorization rules

Après avoir exploré les données et les boxplots, nous décidons d'établir une **règle basée sur les quartiles**, étant donné que les données sont normalisées. Nous proposons donc d'avoir 4 catégories:

Excellent	5% meilleur
Good	de 5% à 25%
Regular	de 25% à 50%
Bad	en dessous de 50%

Après avoir exécuté quelques modèles, nous avons constaté que les **résultats étaient biaisés** en raison des classes déséquilibrées (par exemple, une précision de 98

% pour "Excellent" et une précision de 60 % pour "Bad"). Comme l'erreur était proportionnelle à la quantité de données, après avoir équilibré les classes, nous avons obtenu de meilleurs résultats. Par conséquent, nous avons décidé de le maintenir en tant que partie intégrante de la stratégie principale.

Le rééquilibrage a été effectué en prenant comme référence la quantité de données de la classe "Excellent" afin d'obtenir le même nombre dans les 4 classes. En conséquence, chaque classe compte désormais 176 296 valeurs. Comme cela, **on a obtenu des meilleurs résultats en appliquant les modèles de classifications** qui sont montrées ci-dessous

Classification models

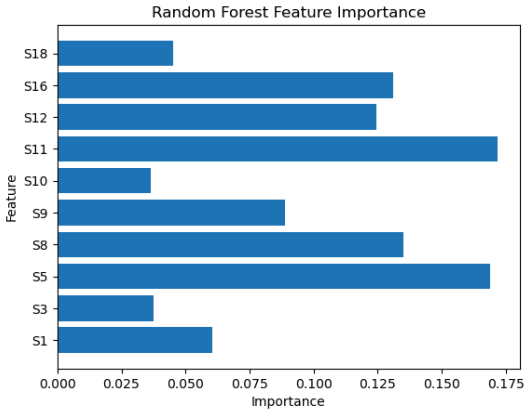
Nous avons mis en œuvre deux modèles de classification différents **: Random Forest et Régression Logistique.** Les valeurs de "Solution Cost", "Cost Category" et de l'instance sont retirées, car les valeurs des "S" ont déjà été normalisées par catégorie et parce que, évidemment, la catégorie de coût a un poids important dans la classification et introduit un biais.

Voici les résultats de l'évaluation de la classification par méthode :

Random Forest					Logistic Regression				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Bad	0,98	0,98	0,98	35402	**Bad**	0,87	0,81	0,84	35402
Excellent	0,97	0,99	0,98	35269	**Excellent**	0,82	0,86	0,84	35269
Good	0,97	0,95	0,96	35040	**Good**	0,69	0,64	0,67	35040
Regular	0,96	0,96	0,96	35326	**Regular**	0,68	0,75	0,71	35326
accuracy			0,97	141037	**accuracy**	-	-	0,76	141037
macro avg	0,97	0,97	0,97	141037	**macro avg**	0,77	0,76	0,76	141037
weighted avg	0,97	0,97	0,97	141037	**weighted avg**	0,77	0,76	0,76	141037

En conséquence, les modèles présentent d'une part une **haute précision dans les 4 catégories**, avec également un indicateur *recall* signalant une performance élevée dans l'attribution des cas positifs. De même, **le modèle *Random Forest* affiche des performances plus élevées et plus cohérentes** que la régression logistique.

Pour explorer la construction du modèle de Random Forest, la propriété “feature importance” indique que les caractéristiques les plus importantes pour le modèle sont S5 et S11, suivies de S8 et S16. Ceci est cohérent avec les résultats de PCA de la première partie.



Model Validation

Pour constater les modèles, **la matrice de confusion** montre la quantité des données bien classifiées et mal classifiées qui est cohérente avec les résultats des mesures du modèle.

Il sont remarquables les erreurs faits dans [Predicted Labels: 0, True Labels: 3] et vice versa, Predicted Labels :3 et True Labels: 0].

		Predicted Labels			
		0	1	2	3
True labels	0	34558	0	12	832
	1	3	34841	407	18
	2	36	1190	33354	460
	3	804	4	648	33870

Cross validation: En effectuant une validation croisée manuelle et en sélectionnant un échantillon aléatoire différent avec une plus grande portion de données, il est possible d'évaluer la généralisation de la méthode avec les résultats suivants

	Precision	recall	f1-score	support
Bad	1	1	1	100523
Excellent	0,99	1	1	100406
Good	0,99	0,99	0,99	100446
Regular	0,99	0,99	0,99	100580
Accuracy			0,99	401955
macro avg	0,99	0,99	0,99	401955
weighted avg	0,99	0,99	0,99	401955

Conclusion :

Nous avons abordé le problème du CVRP, utilisant des solutions générées par une variante d'algorithme génétique. L'objectif final n'était pas de résoudre le problème directement, mais plutôt de développer des modèles de prédiction de la qualité des solutions.

Dans notre approche, nous avons commencé par une phase de statistiques descriptives et de feature engineering, cherchant à comprendre la structure des données et à recoder, transformer, voire créer de nouvelles variables pertinentes. Ensuite, nous avons entrepris un benchmark des méthodes de régression pour prédire le coût des solutions CVRP. En explorant des techniques telles que la régression linéaire, Support Vector Regression, Arbres de régression, Random Forest, Gradient Boosting, et même les Réseaux de Neurones, nous avons cherché à obtenir le meilleur modèle en termes de généralisation sur un ensemble de test.

La phase suivante a été une transformation du problème en un problème de classification. En introduisant une nouvelle variable représentant la qualité de la solution ‘Cost_category’ sous forme catégorielle, nous avons repensé notre approche pour prédire la qualité globale des solutions. Et sur les deux problèmes (régression, classification) nous nous sommes basés sur la validation croisée pour vérifier que nos modèles ont bien des capacités de généralisation.