Algorithme des K plus proches voisin - Projet "Prévoir les survivants du Titanic"

Vous allez travailler sur le jeu de données suivant à télécharger sur Pronote : titanic.csv

Ce jeu de données contient des informations sur une partie des passagers (plus exactement sur 891 passagers) du Titanic. Pour un petit rappel historique, vous pouvez consulter la page Wikipédia consacrée à ce paquebot.
https://fr.wikipedia.org/wiki/Titanic

À faire vous-même 1

Ouvrez le fichier "titanic.csv" à l'aide d'un tableur.
Vous devriez obtenir quelque chose qui ressemble à ceci :

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.925		S
5	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
15	14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275		S
16	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
17	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
18	17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29.125		Q
19	18	1	2	Williams, Mr. Charles Eugene	male		0	0	244373	13		S
20	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31	1	0	345763	18		S
21	20	1	3	Masselmani, Mrs. Fatima	female		0	0	2649	7.225		C
22	21	0	2	Fynney, Mr. Joseph J	male	35	0	0	239865	26		S
23	22	1	2	Beesley, Mr. Lawrence	male	34	0	0	248698	13	D56	S
24	23	1	3	McGowan, Miss. Anna "Annie"	female	15	0	0	330923	8.0292		Q
25	24	1	1	Sloper, Mr. William Thompson	male	28	0	0	113788	35.5	A6	S
26	25	0	3	Palsson, Miss. Torborg Danira	female	8	3	1	349909	21.075		S

Trouvez, en faisant des recherches sur le web, la signification des différents descripteurs : "PassengerId", "Survived", "Pclass"... (ce jeu de données est un grand classique en "machine learning", vous ne devriez donc pas avoir trop de mal à trouver ces informations)

L'idée de ce projet est d'utiliser l'algorithme des k plus proches voisins afin de déterminer si un passager X (non présent dans le fichier titanic.csv) ayant certaines caractéristiques ("PassengerId", "Pclass", "Name", "Sex"...) aurait survécu au naufrage du Titanic.

Pour parvenir à ce résultat, un travail de préparation des données va être nécessaire (tout "data scientist" un peu sérieux vous dira que ce travail de préparation des données est absolument fondamental), vous allez donc devoir passer par quelques étapes décrites ci-après :

- Tous les types de données ne vont pas forcement être pertinents, par exemple, d'après vous, lors du naufrage, le nom du passager a-t-il eu une importance sur le fait qu'il ait ou non survécu ? (Nous ne tiendrons pas compte du fait que certaines personnes aient pu être privilégié au vu de leur nom de famille, sur les 891 passagers présents dans le fichier titanic.csv, ce phénomène est négligeable).

- En analysant le contenu du fichier titanic.csv (par exemple à l'aide d'un tableur), choisissez les descripteurs qui vous paraissent les plus pertinents. Vous effacerez les colonnes qui vous semblent inutiles directement dans le tableur ou dans votre programme python à l'aide de la bibliothèque Pandas (n'hésitez pas à consulter la documentation de Pandas, notamment l'utilisation de la méthode drop).

- Pour certains passagers, il manque des données. Par exemple, l'âge du passager ayant pour "PassengerId" 6 n'est pas renseigné. La solution de facilité serait de supprimer du fichier les passagers ayant des données incomplètes. Il y a d'autres possibilités, essayez d'en proposer au moins une.

- L'utilisation de l'algorithme des k plus proches voisins nous oblige à proscrire les données non numériques. Par exemple, la colonne "Sex" ne peut pas être utilisée telle quelle, l'algorithme n'est pas capable de traiter les "male" et "female". Vous devez donc modifier certaines données. Pour ce faire, je vous conseille d'utiliser la bibliothèque Pandas. Si après quelques recherches vous n'arrivez pas à trouver la solution, n'hésitez pas à demander de l'aide.

- Nous avons vu que pour utiliser le "KNeighborsClassifier" de scikit-learn (scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation), il est nécessaire de séparer les données et le label. Dans notre cas, le label correspond à la colonne "Survived" (c'est ce que nous cherchons à déterminer pour un passager X). Il est donc nécessaire de séparer les données contenues dans la colonne "Survived" des autres données afin d'avoir 2 tableaux : un tableau contenant les données que vous aurez sélectionnées ci-dessus et un tableau contenant uniquement le label "Survived". Ici aussi Pandas vous permet d'effectuer cette séparation relativement facilement, faites quelques recherches (si vous ne trouvez pas au bout d'un certain temps, n'hésitez pas à demander de l'aide).

- Un modèle de machine learning n'a de sens que s'il est possible de l'évaluer. La méthode classique pour effectuer cette évaluation est de séparer les données de départ (les 891 passagers) en 2 groupes : un groupe que l'on nomme souvent "train" qui servira à "entrainer" l'algorithme de machine learning et un groupe "test" qui servira à évaluer la capacité de l'algorithme à prévoir des résultats corrects. Après cette séparation vous devriez avoir les tableaux suivants : X_train (les données d'entrainement), y_train (les labels correspondant aux données d'entrainement), X_test (les données de test) et y_test (les labels correspondant aux données de test). Scikit-learn propose la méthode "train_test_split" qui permet d'effectuer cette séparation très simplement, n'hésitez pas à consulter la documentation de scikit-learn afin d'en savoir plus sur l'utilisation de "train_test_split".

- La méthode "fit" de scikit-learn vous permet de procéder à l'entrainement de l'algorithme (en utilisant les données X_train et y_train).

- La méthode "predict", toujours de scikit-learn, vous permet de tester l'entrainement de votre algorithme. Cette méthode va permettre de générer un tableau que vous nommerez y_pred, y_pred contiendra les labels prévus par l'algorithme pour les données contenues dans X_test.

- Pour évaluer l'entrainement de l'algorithme, il suffira de comparer y_pred et y_test. On pourra obtenir le pourcentage de réponses correctes en utilisant la méthode "metrics.accuracy_score" (voir la documentation de scikit-learn).

- Si vous obtenez un pourcentage correct (supérieur à 65% de bonnes réponses pour y_pred), vous pouvez alors passer à l'écriture de la fonction qui vous permettra de prévoir si votre passager X aurait survécu au naufrage du Titanic.

Ce projet vous permettre de mieux comprendre le travail d'un "data scientist" (un des métiers les plus demandés sur le marché du travail actuellement). N'hésitez pas à demander de l'aide et surtout, bon courage !