Dauphine | PSL UNIVERSITÉ PARIS LAMSADE UMR 7243

Projet Analyse de données

Responsable du cours: Sarra Tajouri
Chargée du cours: Shuhui Wang

January 20, 2025

Présentation du cours

- Volume horaire : 24h (8 TPs de 3h chacun)
- Objectif : Mise en pratique des compétences acquises en analyse de données
- Modalités :
  - Chaque TP doit être rendu sur Moodle.
  - 3 TPs seront corrigés pour constituer la note finale.

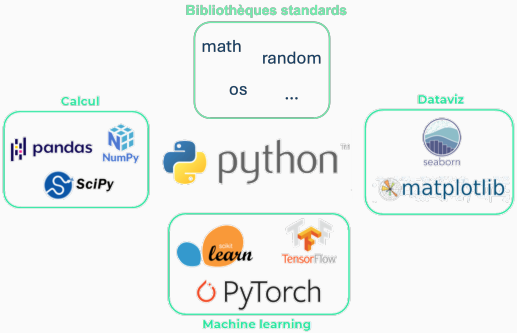
Programme des séances

1. **Séance 1** : Introduction à Pandas
2. **Séance 2** : Régression linéaire et polynomiale
3. **Séance 3** : Régression et optimisation
4. **Séance 4** : Régression multiple
5. **Séance 5** : Prédiction de l'âge
6. **Séance 6** : Analyse en composantes principales
7. **Séance 7** : Factorisation en matrice non-négatives
8. **Séance 8** : Apprentissage par renforcement

Matériel de Yann Chevaleyre

Informations supplémentaires

- Tous les supports et consignes seront disponibles sur Moodle.
- Respectez les deadlines pour la soumission des TPs (avant la séance prochaine)
- Commentez votre code !
- Support pour coder: Jupyter Notebook, VSCode, Colab (désactiver l'IA intégrée)



Librairies Python

- Pour installer une librairie :
  `pip install ...`
- Pour utiliser une librairie :
  `import ...`
  `import ... as ...`
  `from ... import ...`
- Pour accéder à la documentation :
  `help(...)`

Introduction à Pandas

- Pandas est une bibliothèque Python pour l'analyse et la manipulation de données.
- Fournit des structures de données puissantes :
  - **Series** : Vecteurs unidimensionnels avec un index.
  - **DataFrame** : Tableau 2D (similaire à une feuille Excel).

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Commandes utiles pour un DataFrame

- `df.shape` : Affiche la taille du DataFrame (nombre de lignes, colonnes).
- `df.columns` : Liste des noms des colonnes.
- `df.head()` : Affiche les 5 premières lignes (par défaut).
- `df.tail()` : Affiche les 5 dernières lignes (par défaut).
- `df.info()` : Résumé du DataFrame (types de données, mémoire utilisée).
- `df.describe()` : Statistiques descriptives des colonnes numériques.
- `df.dtypes` : Affiche les types des colonnes.
- `df.index` : Affiche les indices (index) du DataFrame.
- `df.isnull().sum()` : Nombre de valeurs manquantes par colonne.
- `df['col'].unique()` : Valeurs uniques d'une colonne.
- `df.nunique()` : Nombre de valeurs uniques pour chaque colonne.
- `df.sort_values(by='col')` : Trie les données selon une colonne.
- `df['col'].value_counts()` : Fréquence des valeurs dans une colonne.

Accéder à une colonne dans un DataFrame

- `dataframe.variable` :
  - Requiert des noms de colonnes sans espaces ni caractères spéciaux.
- `dataframe[['variable']]` :
  - Renvoie la variable sous forme de **DataFrame**.
  - Risque de confusion pour une variable unique.
  - Préférez : `dataframe.loc[:, ['variable']]` pour plus de clarté.
- `dataframe['variable']` :
  - Renvoie la variable sous forme de **Series**.

Accéder aux lignes dans un DataFrame

- `dataframe.loc[index]` :
  - Accède aux lignes par leur **label** (étiquette de l'index).
  - Exemple : `df.loc[3]` pour accéder à la ligne avec l'index 3.
- `dataframe.iloc[position]` :
  - Accède aux lignes par leur **position entière**.
  - Exemple : `df.iloc[3]` pour accéder à la $4^{\text{ème}}$ ligne (ligne 1 indexée à 0).
- `dataframe.loc[start:end]` :
  - Sélectionne un **sous-ensemble de lignes** avec des labels inclusifs.
  - Exemple : `df.loc[2:5]` sélectionne les lignes d'index 2 à 5.
- `dataframe.loc[condition]` :
  - Sélectionne les lignes qui satisfont une **condition**.
  - Exemple : `df.loc[df['Colonne'] > 50]` pour sélectionner les lignes où la colonne a une valeur supérieure à 50.