– TP 1 – Prétraitement, Régression et Classification

Pour ce TP, vous êtes invités à appliquer de manière pratique : (1) Le prétraitement des données, (2) La régression linéaire & polynomiale, (3) La classification : logistique, KNN et arbre de décision, (4) Les métriques d'évaluation, (5) Les visualisations et l'analyse et (6) les interprétations.

Préparation de l'environnement de travail

(1) Ouvrir un notebook ou un éditeur de code dur un dossier séparé (ex : Jupyter, Google Colab, Spyder, …). Assurer que vous êtes dans un environnement de travail, dossier, sépara de vos autres fichiers.

(2) Charger le fichier « **`data.csv`** ». Le fichier sera partagé avec l'énoncé de ce TP.

	age	visits_last_month	time_on_site	is_subscriber	device	engagement	y_true_purchase_amount	y_pred_lin	made_purchase_true	made_purchase_log_pred
1	22	12.0		1	mobile	94.80000000000001	7.9	6.809961340302609	1	1
2	49	4.0	5.1	0	tablet	20.4	0.0	1.1972545566925463	0	0
3	38	7.0	9.1	0	mobile	63.699999999999996	0.0	2.92312997427305	0	0
4	56	10.0	8.1	0	mobile	81.0	10.67	6.498794868133838	1	1
5	59	7.0	4.8	1	mobile	33.6	9.41	5.708198988943405	1	1
6	34	4.0	9.5	1	mobile	38.0	0.0	2.2456273533426367	0	0
7	58	5.0	16.6	0	desktop	83.0	0.0	4.481465334153591	0	1
8	31	11.0	7.0	0	desktop	77.0	0.0	4.673390127995486	0	1
9	43	11.0	6.2	1	mobile	68.2	9.7	7.367176599787496	1	1
10	44	13.0	8.0	1	desktop	104.0	11.06	9.53171900773364	1	1

Étape A : Chargement & exploration initiale

Le fichier « **`data.csv`** » contient des variables numériques et catégorielles, avec quelques valeurs manquantes et bruité.

(3) Charger le fichier dans un DataFrame avec la méthode : **`read_csv()`**, de la bibliothèque **`pandas`**.

(4) Afficher les 5 premières lignes et les descriptions statistiques avec les méthodes : **`df.head()`**, **`df.info()`**, **`df.describe()`**. Vérifier er déduire :

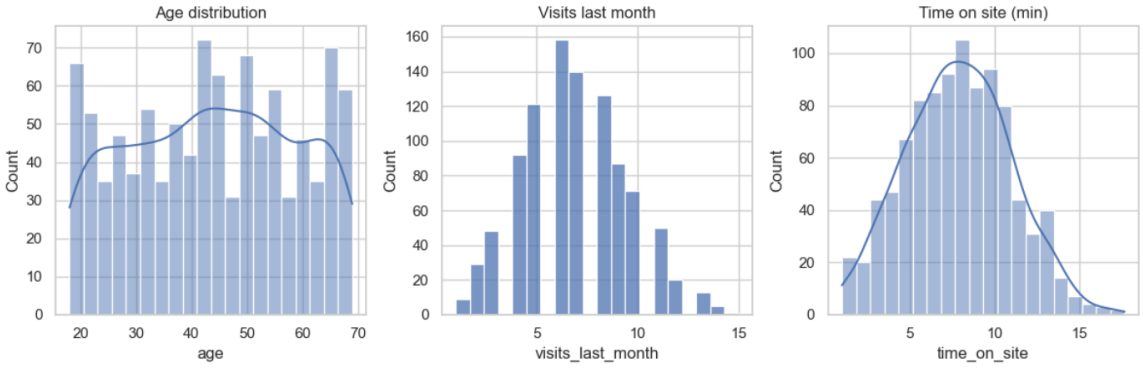
```
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   age                1000 non-null   int64
 1   visits_last_month  970 non-null    float64
 2   time_on_site       970 non-null    float64
 3   is_subscriber      1000 non-null   int64
 4   device             970 non-null    object
 5   made_purchase      1000 non-null   int64
 6   purchase_amount    1000 non-null   float64
dtypes: float64(3), int64(3), object(1)
```

- Types des colonnes : … … … … … … … … … … … …

  …
- Données manquantes : … … … … … … … … … … … …

  …
- Doublons : … … … … … … … … … … … … … … … … …
- Distribution des valeurs : …

(5) Créer les Visualisations ci-dessous, utilisant : ***Histogrammes*** pour chaque variable numérique (**`histplot`**) , ***Diagramme en barres*** pour device (**`countplot`**), ***Heatmap corrélation*** (numériques), ***Pairplot*** (**`seaborn`**) pour inspecter les relations entre les différentes variables.



(6) Détecter valeurs manquantes utilisant la méhode : `df.isna().sum()`.

(7) Quelles variables semblent corrélées avec « ***purchase_amount*** » ? …

(8) « ***made_purchase*** » suit-il la même tendance ? …

Étape B : Prétraitement de données

Pour cette étape, vous êtes invité à suivre les étapes vues durant la séance de cours, et TD de la partie Prétraitement des données. Pour traiter les problèmes vécus dans étape A.

(9) Appliquer des Imputations pour :

- Les variables numériques : `median` pour ***time_on_site***, ***visits_last_month***, expliquer pourquoi ? …
- Les variables Catégorielles : `mode` pour device, expliquer pourquoi ? … … … … … … … … … … … … … … … …

(10) Appliquer l'approche d'Encodage : « ***device*** » → `OneHotEncoder`. NB : is_subscriber laissé tel quel.

(11) Créer une méthode de Feature engineering simple : engagement = visits_last_month * time_on_site.

(12) Appliquer une étape de Scaling (Normalisation/standarisation) : `StandardScaler` pour features utilisées par KNN.

(13) Réaliser les mêmes visualisations de partie A pour justifier les étapes du post-prétraitement : `boxplots` avant/après, distribution de purchase_amount.

(14) Déduire : …

Étape C: Régression (predict purchase_amount)

Pour cette étape, vous êtes invité à entraîner et comparer entre la Régression Linéaire et Polynomiale (deg 2). Utiliser les fonctions prédéfinies de la bibliothèque : `sklearn`.

(15) Séparer les données d'entrainement et testing (80%/20%), utiliser `random_state=42` de la fonction `train_test_split` de la bibiloéthque : `sklearn.model_selection`.

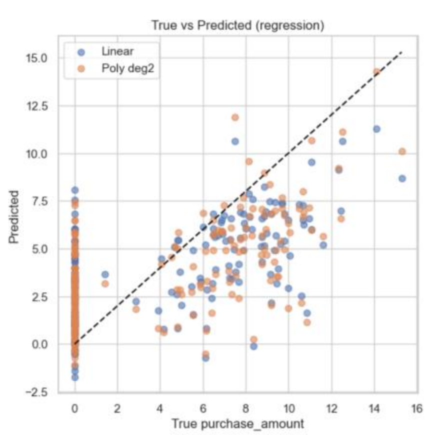
(16) Créer les Features candidates : age, visits_last_month, time_on_site, is_subscriber, one-hot device, engagement.

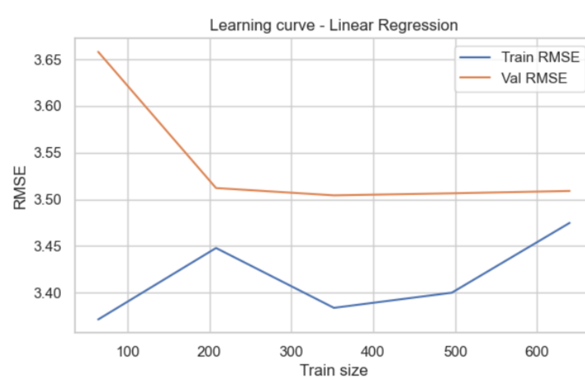
(17) Entraîner les modèles : `LinearRegression` et `PolynomialFeatures(degree=2)`.

(18) Évaluer les deux modèles créer utilisant les données de testing, avec les métriques : MAE, MSE, RMSE, R2.

(19) Créer ces Visualisations utilisant la bibliothèque `matplotlib.pyplot` :

- Scatter y_true vs y_pred (pour chaque modèle) avec ligne y=x.
- Residuals plot (résidu vs préd).
- Learning curve (train/val RMSE en fonction de la taille d'échantillon) — pour voir étudier sur ou underfitting.





(20) Choisir le modèle le plus adapté a ce jeu de données. Justifier et enregistrer ce modèle pour les prédictions

(21) Le polynôme réduit-il erreur ? y a-t-il surapprentissage ? …

Étape D: Classification (predict made_purchase)

Dans cette partie, vous êtes invité à créer des modèles de classification avec un entrainement de Logistic Regression, KNN, Decision Tree. Utiliser les fonctions prédéfinies de la bibliothèque : `sklearn`.

(22) Choisir les features appropriés pour cette classification : …

… …

(23) Créer les partition train/test split identique à la problématique de régression.

(24) Créer les modèles de classification avec l'Entraînement & prédiction pour chaque modèle :

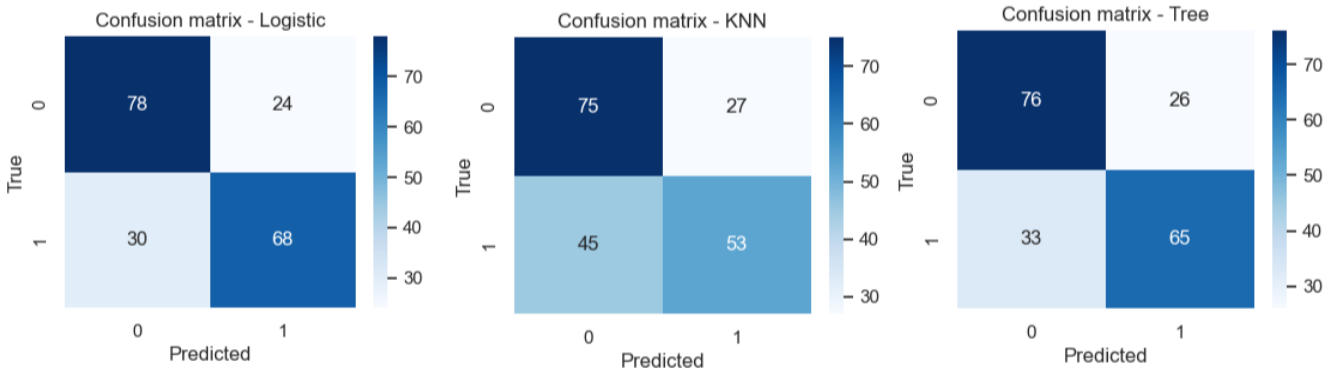
- LogisticRegression (solver='lbfgs'),
- KNeighborsClassifier (avce les valeurs k=3,5,7, pour comparer entre ces valeurs),
- DecisionTreeClassifier (max_depth=4 et tester)

(25) Évaluer chaque modèle avec les métrique de la bibliothèque `sklearn.metrics :`

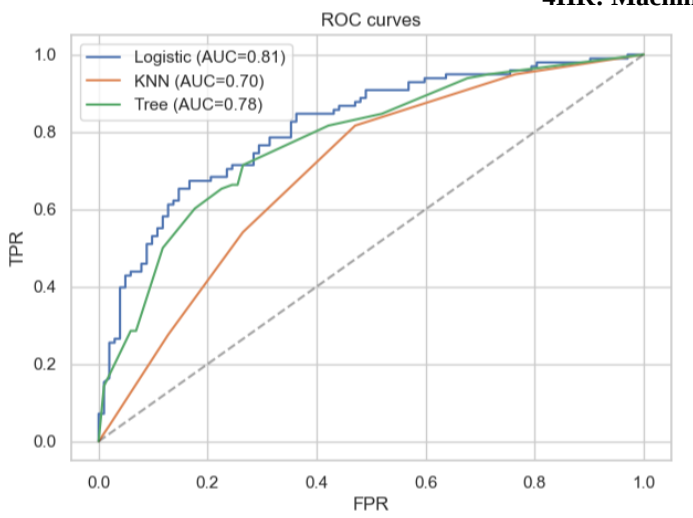
- accuracy, precision, recall, f1_score
- matrice de confusion (heatmap)
- ROC curve + AUC
- Precision-Recall curve
- Calibration curve (optionnel, utile pour logistic)

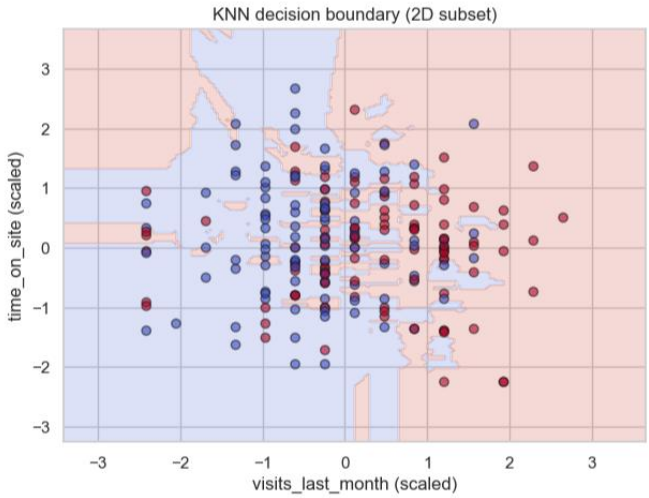
(26) Créer ces visualisations supplémentaires :

- Importance des features (bar chart) pour l'arbre.
- Courbe performance (accuracy/F1) en fonction de k pour KNN.
- Decision boundary visual (sur un sous-ensemble 2D : visits_last_month vs time_on_site)



(27) Comparer modèles et justifier choix (préciser tradeoffs: sensibilité vs précision).





(28) Quel modèle minimiserait les faux négatifs (FN) si l'entreprise veut capturer acheteurs potentiels ? (Réponse: privilégier rappel)

(29) Quelle conséquence d'un dataset non normalisé sur KNN ?

Étape E : Synthèse & Rapport

(30) Résumer résultats (tableau comparatif des métriques).

(31) Visualisations clés à inclure dans le rapport (scatter préd vs vrai, ROC, confusion matrices, feature importances).

(32) Recommandations business : quel modèle déployer pour régression et pour classification ? pourquoi ?