



Rapport de fin de formation

Thème : Social Network Ads

Réalisé par :

- ✓ Messaadi Chaima et Zeraoulia Oumaima
- ✓ **Option** : Licence ingénierie des systèmes d'information et des logiciels

Encadré par :

- ✓ LAGHOUB Nassim
- ✓ **Qualité** : Ingénieur d'état en informatique

Table des matières

Remerciements	6
Introduction :	7
Présentation du thème :	7
Objectifs :	7
Etude de l'existant :	7
Description du dataset :	9
Exploration de données :	12
Construction du modèle	19
Evaluation du modèle :	25
Tests :	26
Conclusion :	29
Références :	30

Table des figures

Figure 1:Importation des packages.....	9
Figure 2:Lecteur de dataset.....	10
Figure 3:Afficher les 5 première ligne de dataset.....	10
Figure 4:Dimensions de dataset.....	10
Figure 5: Affichage des colonnes de dataset.....	11
Figure 6: Description de dataset.....	11
Figure 7:Vérification si la personne a acheté ou non?.....	12
Figure8:Combien d’homme et de femme avons-nous?.....	13
Figure 9:Un cercle graphique qui représente le pourcentage des genres....	13
Figure 10:Un histogramme des âges.....	14
Figure 11:Un histogramme des salaires estimé.....	15
Figure 12:Un histogramme représente si la personne intéresse à l'acheter ou non en fonction de l'âge.....	16
Figure 13:Un histogramme représente si la personne intéresse à l'acheter ou non en fonction de salaire.....	17
Figure 14:Un nuage de point en fonction des deux attributs Age et EstimatedSalary.....	18
Figure 15:Remplacement les lignes de la colonne Gender par des valeurs numérique 1 ou 2 (1 pour Mâle et 2 pour femme).....	19
Figure 16: La Matrice de corrélation.....	20
Figure 17:Suppression les colonnes inutile de dataset.....	21
Figure 18:Affectation à une variables x les attributs Gender Age EstimatedSalary.....	21
Figure 19:Affectation à une variable y l'attribut Purchased.....	22
Figure 20:Découpage le dataset en 2 données d'entraînements et données de tests.....	23

Figure 21:Affichage la dimension de les données d'entraînements et données de tests.....	23
Figure 22:la courbe d'apprentissage.....	24
Figure 23:Instancier le modèle.....	24
Figure 24:Entraînement le modèle avec les paramètre x_train et y_train...	25
Figure 25:Le coefficient de détermination.....	25
Figure 26:La matrice de confusion.....	25
Figure 27:La matrice de confusion pour comparer les valeurs de test prévues et réelles.....	27
Figure 28:La précision.....	28

Liste des tableaux

Tableau 1:Définition les variables de dataset.....	11
Tableau 2: Explication les variables obtenu lors l'exécution de la fonction describe().....	12

Remerciements

Nous tenons à remercier sincèrement Monsieur Ens LAGHOUB en tant que professeur encadrant de notre formation ,pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter nos réflexions

Ainsi nous le remercions pour l'aide et tout le temps qu'il a bien voulu nous consacrer.

Et enfin, nous remercions tous ceux qui ont contribué de près ou de loin pour réussir cette formation .

Introduction :

L'ensemble de données contient des détails sur l'achat d'un produit basé sur des publicités sur les réseaux sociaux en fonction du sexe, de l'âge ,salaire estimé. La classe ciblé appartient à deux catégories distinctes: acheteur ou non-acheteur. Dans ce projet la classification est faite si une personne achètera un produit affiché sur une publicité de réseau social en utilisant K voisin le plus proche(nearest neighbor).

Présentation du thème :

La publicité sur les réseaux sociaux: est un groupe de termes utilisés pour décrire les formes de publicité en ligne / marketing numérique axées sur les services de réseaux sociaux . L'un des principaux avantages de ce type de publicité est que les annonceurs peuvent tirer parti des informations démographiques des utilisateurs et cibler leurs annonces de manière appropriée.

Objectifs :

Dans ce projet on a créé une application qui permet de classifier les publications sur les réseaux sociaux pour les gens ciblés.

Etude de l'existant :

Il existe un algorithme de Facebook dicte qui voit vos publications et qui ne les voit pas ,il s'appelle Edgerank.

Un autre l'algorithme de Facebook se nomme le « News Feed Algorithm » qui détermine:

- Quand une publication doit être diffusée ou pas dans le fil d'actualités d'un utilisateur.
- Si oui, dans quelle position classer cette publication dans le fil d'actualités.

Il existe aussi un exemple d'instagram et sa chasse aux commentaires: Sur n'importe quel réseau social, vous pouvez recevoir des commentaires haineux, offensants, racistes, etc... Mais avec l'IA d'Instagram, vos commentaires seront

analysés pour effacer ceux qui sont offensants. Avant la publication d'un commentaire, vous pourrez peut-être même apercevoir un message vous demandant si vous êtes sûr de publier ce que vous avez écrit.

Il existe un algorithme de modération YouTube.

l' exemple d'application "JUMIA " qui nous annoncent des publicités sur des produits qui nous intéressent.

Description du dataset :

- **Les bibliothèques :**

Tout d'abord importons tous les bibliothèques python nécessaires que nous utilisons :

Pandas : permettent la manipulation et l'analyse des données.

numpy : est le package fondamental pour le calcul scientifique avec Python.

matplotlib : est une célèbre bibliothèque pour tracer des graphiques en Python.

sklearn.linear_model : est une classe du module sklearn qui contient de différentes fonctions afin d'effectuer un apprentissage automatique avec des modèles linéaires.

SGDREGRESSOR : on peut spécifier une fonction de perte et utiliser l'algorithme de gradient descent stochastique SGD pour s'adapter.

Seaborn : est une bibliothèque qui permet la création des graphiques statistiques en Python. Elle est basée sur Matplotlib.

sklearn.neighbors : on a utilisé pour entraîner le modèle(classification)(à partir de celle-ci, nous apporterons le classificateur KNN).

sklearn.metrics : on a utilisé pour afficher la matrice de confusion.

sklearn.model_selection : on a utilisé pour découper les données disponible en 2 parties : données d'entraînements et données de tests.

Importation des packages

```
Entrée [203]: import pandas as pd # Importation du package pandas
import matplotlib.pyplot as plt # Importation du package matplotlib.pyplot
import numpy as np # Importation du package numpy
from sklearn.linear_model import SGDRegressor # Importation SGDRegressor à partir de la bibliothèque sklearn.linear_model
import seaborn as sns # Importation du package seaborn
from sklearn.neighbors import KNeighborsClassifier # Importation KNeighborsClassifier à partir de la bibliothèque sklearn.neighbors
from sklearn.metrics import confusion_matrix # Importation confusion_matrix à partir de la bibliothèque sklearn.metrics
from sklearn.metrics import accuracy_score # Importation accuracy_score à partir de la bibliothèque sklearn.metrics
from sklearn.model_selection import train_test_split # Importation train_test_split à partir de la bibliothèque sklearn.model_sel
```

Figure 1: Importation des packages

- **Chargement de dataset :**

lecture de la bdd

```
Entrée [3]: data = pd.read_csv('Social_Network_Ads.csv') # Lecture des données à partir d'un fichier
data #Appeler data pour l'afficher
```

```
Out[3]:
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

Figure 2:Lecteur de dataset

- Affichage les 5 premières ligne de dataset :

Afficher les 5 premières ligne du dataframe

```
Entrée [8]: data.head()
```

```
Out[8]:
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

Figure 3:Afficher les 5 première ligne de dataset

-La dimension de notre dataset est : 400 lignes et 5 colonnes.

dimensions

```
Entrée [6]: data.shape #Dimensions du dataset
```

```
Out[6]: (400, 5)
```

Figure 4: Dimensions de dataset

Les colonnes de notre dataset sont : « User ID, Gender , Age, EstimatedSalary, Purchased ».

Colonnes

```
Entrée [7]: data.columns # Afficher Les colonnes
Out[7]: Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'], dtype='object')
```

Figure 5:Affichage des colonnes de dataset

Nom de variable	Définition
User ID	Identifiant d'utilisateur
Gender	Le genre (femme,homme)
Age	l'âge de l'utilisateur
EstimatedSalary	Le revenu mensuel (salaire)
Purchased	Acheteur ou non acheteur (0=Non, 1=Oui)

Tableau 1:Définition les variables de dataset

-Examinons maintenant quelques-unes des informations tirées des données

description de dataset

```
Entrée [9]: data.describe()
Out[9]:
```

	User ID	Age	EstimatedSalary	Purchased
count	4.000000e+02	400.000000	400.000000	400.000000
mean	1.569154e+07	37.655000	69742.500000	0.357500
std	7.165832e+04	10.482877	34096.960282	0.479864
min	1.556669e+07	18.000000	15000.000000	0.000000
25%	1.562676e+07	29.750000	43000.000000	0.000000
50%	1.569434e+07	37.000000	70000.000000	0.000000
75%	1.575036e+07	46.000000	88000.000000	1.000000
max	1.581524e+07	60.000000	150000.000000	1.000000

Figure 6:Description de dataset

count	compter(le nombre des valeurs disponible de chaque colonne)
mean	La moyenne
std	écart-type
min	minimum valeur de chaque colonne
25%	25 centile
50%	50 centile
75%	75 centile
max	maximum valeur de chaque colonne

Tableau 2: Explication les variables obtenu lors l'exécution de la fonction describe()

Exploration de données :

Explorons maintenant certains des modèles importants dans l'ensemble de données. La première chose qu'on veut explorer est le nombre des personnes acheteurs ou non ; on exécute la commande suivante :

verification si la personne achete ou non ?

```
Entrée [10]: data['Purchased'].value_counts()
Out[10]: 0    257
         1    143
         Name: Purchased, dtype: int64
```

Figure 7:Vérification si la personne acheté ou non

D'après le résultat d'exécution on a 257 personnes qui n'ont pas acheté et 143 personnes qui ont acheté .

Explorer le cercle graphique qui représente le pourcentage des Genres.

Pour cela on affiche le nombre des femmes et hommes dans notre dataset. d'où on a 204 femmes et 196 hommes.

Combien de mâles et de femmes avons-nous ?

```
Entrée [13]: data['Gender'].value_counts() '''Il ya 204 femmes et 196 Mâles'''
Out[13]: Female    204
        Male      196
        Name: Gender, dtype: int64
```

Figure8: Combien d'homme et de femme avons-nous?

Voici le cercle graphique :

un cercle graphique qui représente le pourcentage des Genres

```
Entrée [16]: # Avec Le pourcentage on a 51% Femelles et 49% mâle
plt.figure(figsize=(12,8))
a=['Female','Male']
b=[204,196]
plt.pie(b,labels=a,shadow=True,autopct='%1.1f%%', startangle=90)
plt.legend

Out[16]: <function matplotlib.pyplot.legend(*args, **kwargs)>
```

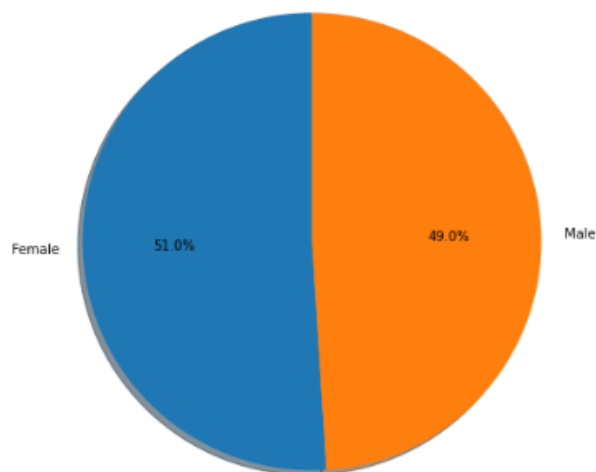


Figure 9: Un cercle graphique qui représente le pourcentage des genres
D'après le résultat d'exécution on a 51% femmes et 49% hommes.

Effectuation d' un histogramme des âges ,

un histogramme des âges

```
Entrée [101]: plt.figure(figsize=(15, 10))
               plt.title("Un histogramme représente le nombre d'âges des personnes")
               plt.xlabel('Age')
               plt.ylabel('Count')
               data['Age'].hist()

Out[101]: <AxesSubplot:title={'center':'Un histogramme représente le nombre d'âges des personnes'}, xlabel='Age', ylabel='Count'>
```

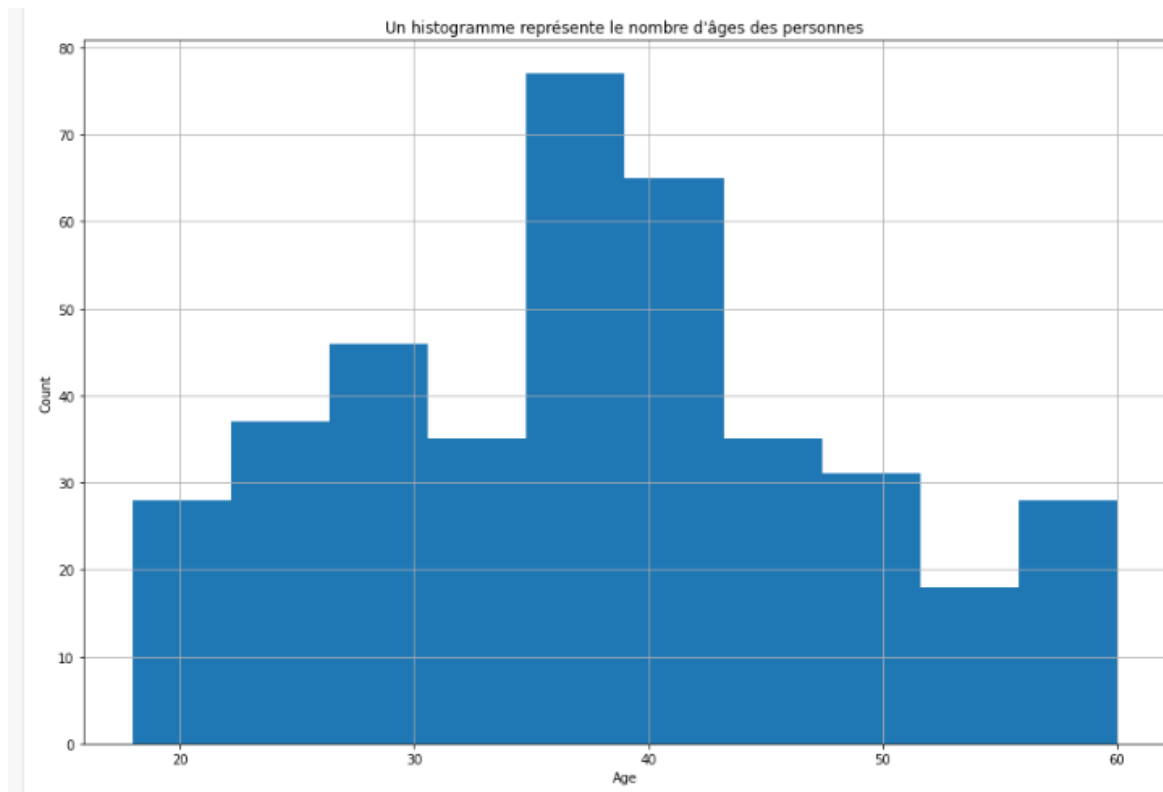


Figure 10:Un histogramme des âges

On note qu'il y a environ 77 personnes âgées entre 35 et 37 ans (la plus haute catégorie).

Il y a environ 18 personnes âgées entre 52 et 56 ans (la plus basse catégorie).

Effectuation d'un histogramme des salaire Estimé

un histogramme des Salaire Estimé

un histogramme des Salaire Estimé

```
Entrée [102]: plt.figure(figsize=(15, 10))
plt.title("Un histogramme représente le nombre des salaire estimé")
plt.xlabel('Estimated Salary')
plt.ylabel('Count')
data['EstimatedSalary'].hist()

Out[102]: <AxesSubplot:title={'center':'Un histogramme représente le nombre des salaire estimé'}, xlabel='Estimated Salary', ylabel='Count'>
```

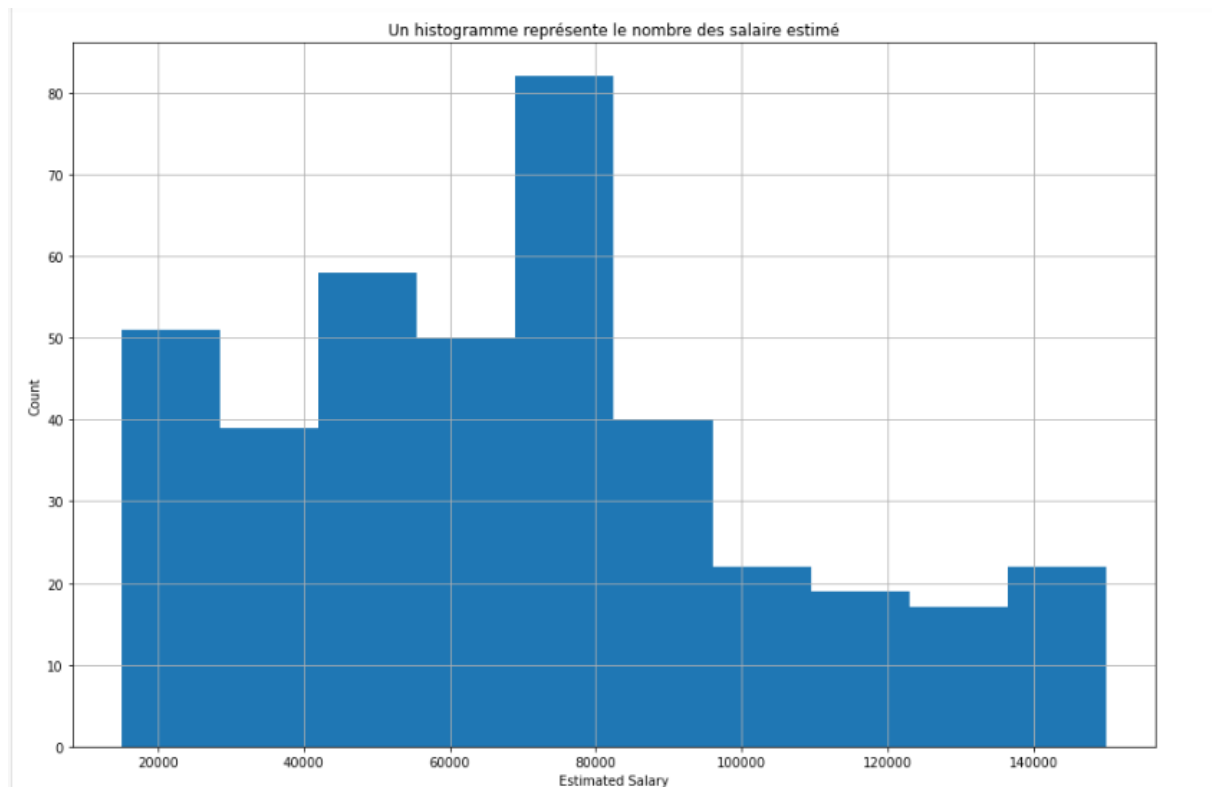


Figure 11:Un histogramme des salaires estimé

On note qu'il y a environ 82 personnes avec un salaire mensuel entre 70000 et 80000.(la plus haute catégorie)

Il y a environ 17 personnes avec son salaire mensuel entre 120010 et 138999.
(la plus basse catégorie).

Dans cet étape on veut élaborer l'âge des personnes qui ont répondu aux publicités sur les réseaux sociaux et acheté le produit :

```
Entrée [7]: plt.figure(figsize=(15, 10))
plt.title('Produit acheté par des personnes via le marketing des médias sociaux')
sns.histplot(x='Age', hue='Purchased', data=data)
plt.show()
```

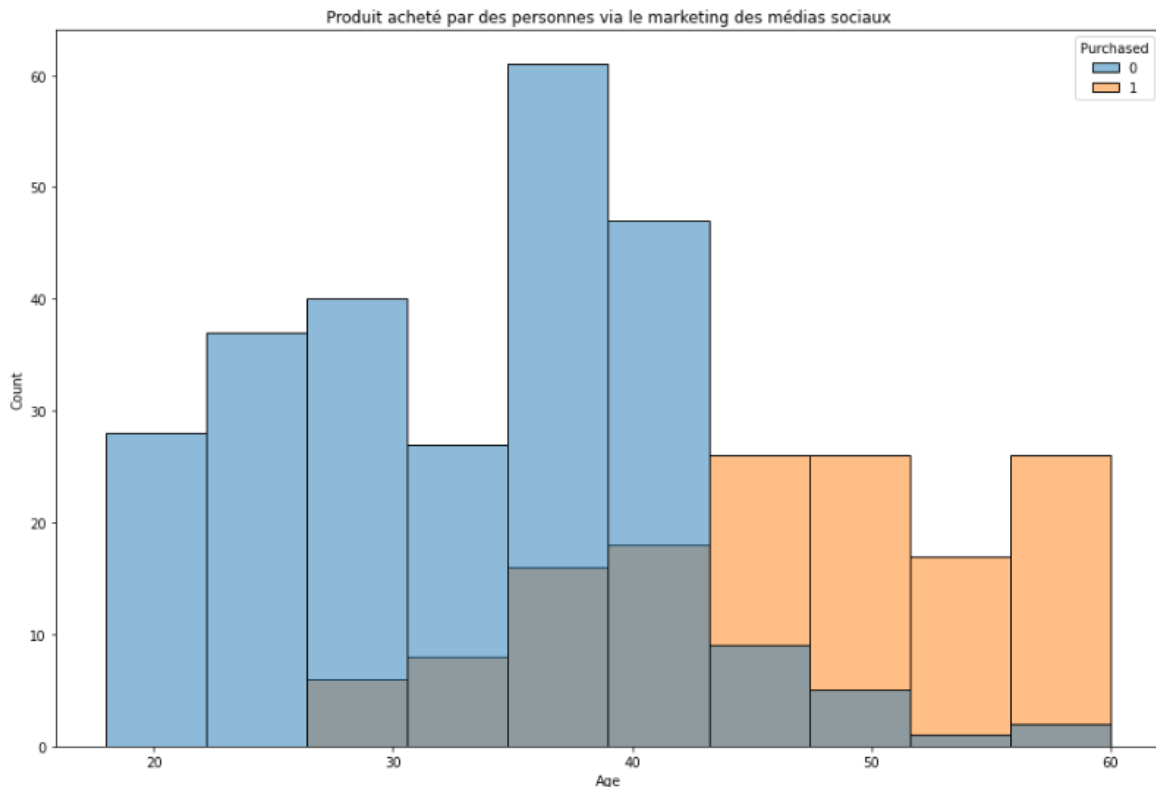


Figure 12:Un histogramme représente si le personne s'intéresse à l'achat ou non en fonction de l'âge.

La visualisation ci-dessus montre que les personnes âgées de plus de 45 ans parmi le public cible sont plus intéressées par l'achat du produit.

Examinons maintenant le groupe de revenus des personnes qui ont répondu aux publicités sur les réseaux sociaux et acheté le produit :

```
Entrée [8]: plt.figure(figsize=(15, 10))
plt.title('Produit acheté par les gens en fonction de leurs revenus ')
sns.histplot(x='EstimatedSalary', hue='Purchased', data=data)
plt.show()
```

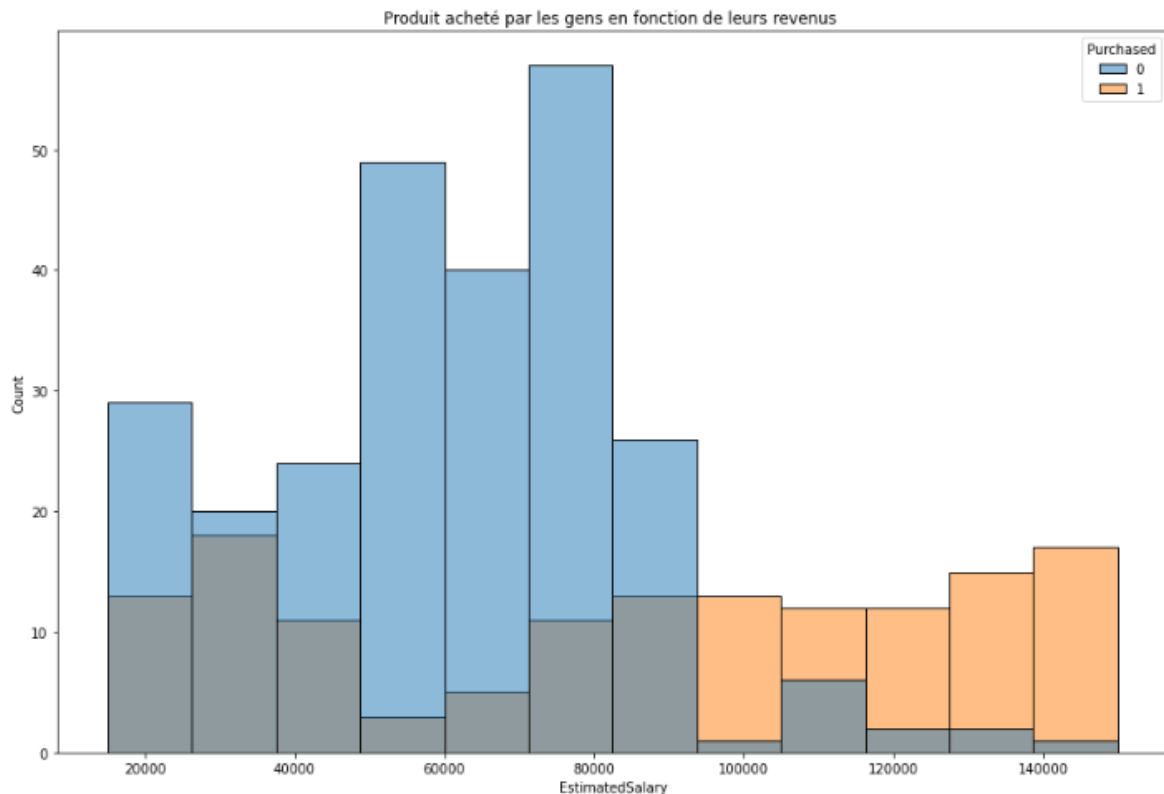



Figure 13: Un histogramme représente si la personne s'intéresse à l'achat ou non en fonction du salaire.

La visualisation ci-dessus montre que les personnes ayant un revenu mensuel supérieur à 90 000 parmi le public ciblé sont plus intéressées par l'achat du produit.

-Nuage de point en fonction des deux attributs Age et EstimatedSalary:

La Nuage de point en fonction des deux attributs Age et EstimatedSalary

```
Entrée [108]: plt.figure(figsize=(15, 10))
plt.scatter(data['Age'], data['EstimatedSalary'], c=data['Purchased'])
plt.colorbar()
```

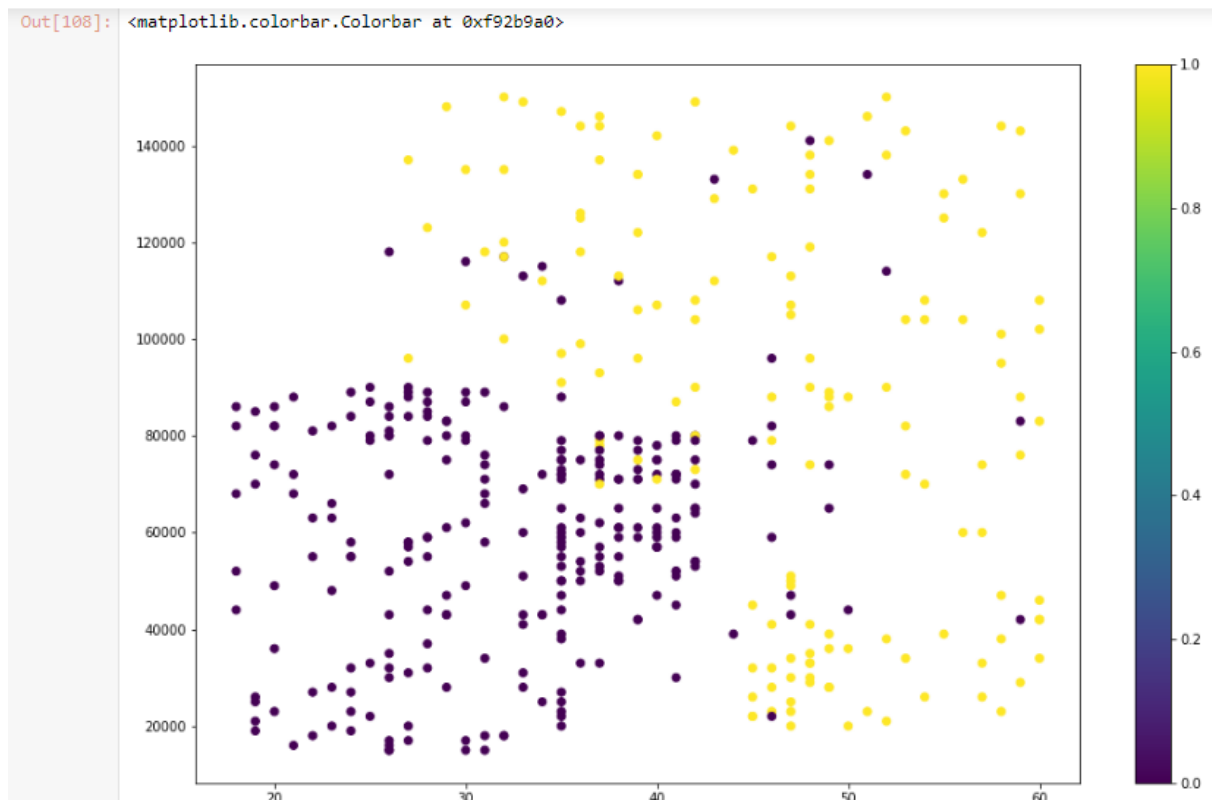


Figure 14:Un nuage de point en fonction des deux attributs Age et EstimatedSalary

On observe que les personnes âgées de plus de 45 ans et ayant un revenu mensuel supérieur à 90 000 ou entre 20000 et 30000 parmi le public cible sont plus intéressées par l'achat du produit .

Construction du modèle

Entraînons maintenant un modèle pour classer les publicités sur les réseaux sociaux. Tout d'abord, on remplace les lignes de la colonne Gender par des valeurs numérique 1 ou 2 (1 pour Mâle et 2 pour femme).

Remplacement les lignes de la colonne Gender par des valeurs numérique 1 ou 2 (1 pour Mâle et 2 pour femme)

Entrée [217]: `data['Gender'].replace(['Male','Female'],[1,2],inplace=True)`

Figure 15: Remplacement des lignes de la colonne Gender

-Affichage du dataset pour vérifier que le remplacement à été affecté .

Entrée [218]: `data`

Out[218]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	1	19	19000	0
1	15810944	1	35	20000	0
2	15668575	2	26	43000	0
3	15603246	2	27	57000	0
4	15804002	1	19	76000	0
...
395	15691863	2	46	41000	1
396	15706071	1	51	23000	1
397	15654296	2	50	20000	1
398	15755018	1	36	33000	0
399	15594041	2	49	36000	1

400 rows × 5 columns

Ensuite, on a élaboré la matrice de corrélation pour voir les variables importantes afin d'entraîner le modèle.

La figure suivante montre la matrice de corrélation:

matrice de corrélation

```
Entrée [219]: corrmat = data.corr()
plt.subplots(figsize=(10,6))
sns.heatmap(corrmat,annot=True,linewidth=0.5,fmt=".2f", cmap="viridis")
```

Out[219]: <AxesSubplot:>

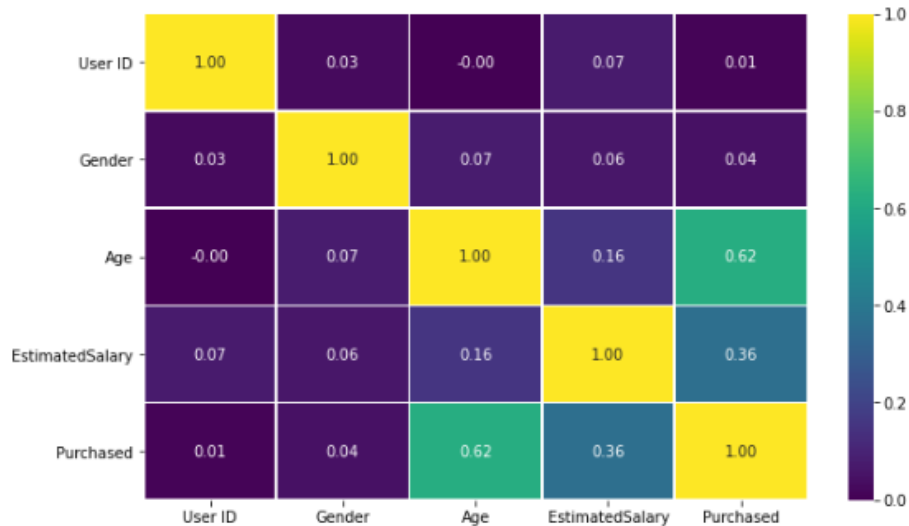


Figure 16: La Matrice de corrélation

Il existe une corrélation négative entre Age et User ID et des corrélations faibles entre User ID et les autres variables et c'est clair dès le départ que le variable User ID n'est pas important pour entraîner le modèle.

Par contre il existe une corrélation pas mal entre les autres variables (Age, Gender, Estimated Salary et Purchased)

Donc on a besoin de ces variables (Age, Gender, EstimatedSalary et Purchased) pour entraîner le modèle sachant que le variable Purchased c'est la sortie de notre modèle.

- Ensuite on supprime toutes les colonnes inutiles de notre dataset:

On a pas besoin de la colonne User ID , on la supprime avec la commande suivante:

Supprimer les colonnes inutiles de dataset

Entrée [5]: `data=data.drop(['User ID'],axis=1) #suppression la colonne User ID`

Entrée [6]: `data`

Out[6]:

	Gender	Age	EstimatedSalary	Purchased
0	Male	19	19000	0
1	Male	35	20000	0
2	Female	26	43000	0
3	Female	27	57000	0
4	Male	19	76000	0
...
395	Female	46	41000	1
396	Male	51	23000	1
397	Female	50	20000	1
398	Male	36	33000	0
399	Female	49	36000	1

400 rows x 4 columns

Figure 17: Suppression les colonnes inutile de dataset

on affecte à une variable x les attributs Gender Age EstimatedSalary avec la commande suivant:

Affecter à une variables X les Gender Age EstimatedSalary

Entrée [9]: `x=data[['Gender','Age','EstimatedSalary']]`

Figure 18: Affectation à une variables x des attributs

-Affichage de la variable x pour s'assurer que l' affectation est faite :

Entrée [10]: `x`

Out[10]:

	Gender	Age	EstimatedSalary
0	Male	19	19000
1	Male	35	20000
2	Female	26	43000
3	Female	27	57000
4	Male	19	76000
...
395	Female	46	41000
396	Male	51	23000
397	Female	50	20000
398	Male	36	33000
399	Female	49	36000

400 rows × 3 columns

- On affecte à une variable `y` l'attribut `Purchased` avec la commande suivant:

Affecter à une variable `Y` l'attribut `Purchased`

Entrée [11]: `y=data[['Purchased']]`

Figure 19: Affectation à une variable `y` l'attribut `Purchased`.

Affichage de la variable `y` :

```
Entrée [12]: y
```

```
Out[12]:
```

	Purchased
0	0
1	0
2	0
3	0
4	0
...	...
395	1
396	1
397	1
398	0
399	1

400 rows × 1 columns

On a séparé le dataset en 2 parties : données d'entraînements et données de tests .

On fournit une taille de test de 0,1 , ce qui signifie que notre échantillon d'entraînement contient 360 lignes d'entraînement et que l'échantillon de test contient 40 lignes de tests.

```
Entrée [241]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.1)
```

Figure 20:Découpage le dataset en 2 données d'entraînements et données de tests

```
Entrée [242]: print('Train set: ',x_train.shape)
               print('Test set: ',x_test.shape)
```

```
Train set: (360, 3)
Test set: (40, 3)
```

Figure 21:Affichage la dimension de les données d'entraînements et données de tests

Nous devons maintenant entraîner le modèle K Nearest Neighbor avec les données d'entraînement pour cela on a fait la courbe d'apprentissage, Grâce à

l'algorithme KNN on peut choisir le K pour lequel la classification sera la meilleure.

La figure suivante montre la courbe d'apprentissage:

La courbe d'apprentissage

```
Entrée [243]: K=[]  
Score = []  
for i in range (1,150):  
    K.append(i)  
    model=KNeighborsClassifier(n_neighbors = i)  
    model.fit(x_train,y_train)  
    Score.append(model.score(x_train,y_train)*100)  
plt.plot(K,Score)
```

Out[243]: [<matplotlib.lines.Line2D at 0xe61a730>]

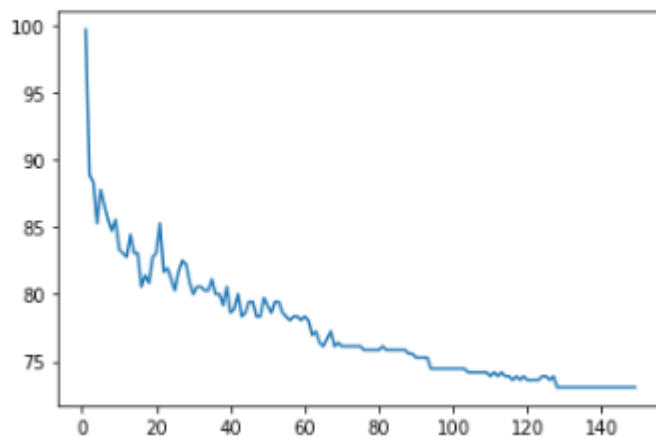


Figure 22:la courbe d'apprentissage

Nous pouvons voir que la meilleure valeur de K est égale à 1.

Ensuite, on instancier maintenant le modèle avec le paramètre K=1.

Instancier le model

```
Entrée [64]: model = KNeighborsClassifier(n_neighbors=1)
```

Figure 23: Instancier le modèle

Entraînons maintenant notre modèle avec les paramètres x_train et y_train, en utilisant la fonction fit.

Entraîner votre modèle avec les paramètres x_train et y_train

```
Entrée [245]: model.fit(x_train,y_train)
```

D:\Nouveau dossier (2)\lib\site-packages\sklearn\neighbors_classification.py:179: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```
return self._fit(X, y)
```

Out[245]: KNeighborsClassifier(n_neighbors=1)

Figure 24: Entraînement le modèle avec les paramètre x_train et y_train**Evaluation du modèle :**

Pour évaluer notre modèle, on calcule le coefficient de détermination de notre Dataset, pour cela on utilise la fonction score du modèle défini précédemment, il est utilisé pour évaluer les performances du modèle.

```
Entrée [246]: print('Coeff R2 =', model.score(x_train, y_train)) #calcule le coefficient de détermination
```

Coeff R2 = 0.9972222222222222

Figure 25:Le coefficient de détermination

Le coefficient de détermination est égale à 0.9972222222222222 et ce résultat est bon.

Etablir la matrice de confusion :

La matrice de confusion, appelée également matrice d'erreur, est un tableau qui présente différentes prévisions et résultats de tests, en les comparant avec des valeurs réelles. Ces matrices sont utilisées dans les modèles d'apprentissage automatique (machine learning) et d'autres applications de l'intelligence artificielle.

La matrice de confusion

```
Entrée [247]: cm = confusion_matrix(model.predict(x_train),y_train)
              print(cm)
```

```
[[233  1]
 [  0 126]]
```

Figure 26:La matrice de confusion

D'après l'observation de la matrice de confusion :

233 signifie que 233 résultats positifs sont effectivement positifs (233 personnes sont intéressées par acheter et réellement ils sont intéressées par acheter).

126 signifie que 126 résultats négatifs sont effectivement négatifs (126 personnes ne sont pas intéressées par acheter et réellement il y a 126 personnes qui ne sont pas intéressées par acheter).

1 signifie qu'il y a une personne intéressée par acheter mais réellement il y a une personne qui ne s'intéresse pas à acheter.

0 signifie qu'il y a 0 personne qui n'est pas intéressé par acheter mais réellement il y a 0 personne intéressée par acheter.

Tests :

Notre modèle est créé, nous devons maintenant prédire la sortie de l'ensemble de test.

Test

```
Entrée [248]: y_pred = model.predict(x_test)
```

Comparaison de la valeur vraie et prédite :

Entrée [253]:	y_test																																																																																		
Out[253]:																																																																																			
	<table> <thead> <tr> <th></th><th>Purchased</th></tr> </thead> <tbody> <tr><td>37</td><td>0</td></tr> <tr><td>180</td><td>0</td></tr> <tr><td>307</td><td>1</td></tr> <tr><td>239</td><td>1</td></tr> <tr><td>196</td><td>0</td></tr> <tr><td>219</td><td>1</td></tr> <tr><td>362</td><td>1</td></tr> <tr><td>50</td><td>0</td></tr> <tr><td>272</td><td>1</td></tr> <tr><td>226</td><td>1</td></tr> <tr><td>128</td><td>0</td></tr> <tr><td>89</td><td>0</td></tr> <tr><td>263</td><td>0</td></tr> <tr><td>340</td><td>1</td></tr> <tr><td>61</td><td>0</td></tr> <tr><td>287</td><td>1</td></tr> <tr><td>181</td><td>0</td></tr> <tr><td>355</td><td>1</td></tr> <tr><td>313</td><td>1</td></tr> <tr><td>144</td><td>0</td></tr> <tr><td>25</td><td>1</td></tr> <tr><td>99</td><td>0</td></tr> <tr><td>51</td><td>0</td></tr> <tr><td>376</td><td>0</td></tr> <tr><td>374</td><td>0</td></tr> <tr><td>357</td><td>0</td></tr> <tr><td>47</td><td>0</td></tr> <tr><td>368</td><td>0</td></tr> <tr><td>319</td><td>0</td></tr> <tr><td>273</td><td>1</td></tr> <tr><td>199</td><td>0</td></tr> <tr><td>90</td><td>0</td></tr> <tr><td>256</td><td>0</td></tr> <tr><td>324</td><td>1</td></tr> <tr><td>197</td><td>0</td></tr> <tr><td>232</td><td>1</td></tr> <tr><td>252</td><td>1</td></tr> <tr><td>188</td><td>0</td></tr> <tr><td>135</td><td>0</td></tr> <tr><td>318</td><td>1</td></tr> </tbody> </table>		Purchased	37	0	180	0	307	1	239	1	196	0	219	1	362	1	50	0	272	1	226	1	128	0	89	0	263	0	340	1	61	0	287	1	181	0	355	1	313	1	144	0	25	1	99	0	51	0	376	0	374	0	357	0	47	0	368	0	319	0	273	1	199	0	90	0	256	0	324	1	197	0	232	1	252	1	188	0	135	0	318	1
	Purchased																																																																																		
37	0																																																																																		
180	0																																																																																		
307	1																																																																																		
239	1																																																																																		
196	0																																																																																		
219	1																																																																																		
362	1																																																																																		
50	0																																																																																		
272	1																																																																																		
226	1																																																																																		
128	0																																																																																		
89	0																																																																																		
263	0																																																																																		
340	1																																																																																		
61	0																																																																																		
287	1																																																																																		
181	0																																																																																		
355	1																																																																																		
313	1																																																																																		
144	0																																																																																		
25	1																																																																																		
99	0																																																																																		
51	0																																																																																		
376	0																																																																																		
374	0																																																																																		
357	0																																																																																		
47	0																																																																																		
368	0																																																																																		
319	0																																																																																		
273	1																																																																																		
199	0																																																																																		
90	0																																																																																		
256	0																																																																																		
324	1																																																																																		
197	0																																																																																		
232	1																																																																																		
252	1																																																																																		
188	0																																																																																		
135	0																																																																																		
318	1																																																																																		

Entrée [249]: y_pred

Out[249]: array([0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1], dtype=int64)

Nous pouvons évaluer notre modèle à l'aide de la matrice de confusion et du score de précision en comparant les valeurs de test prévues et réelles

La matrice de confusion

Entrée [250]: `cm = confusion_matrix(y_test, y_pred)`
cm

Out[250]: array([[21, 3],
[3, 13]], dtype=int64)

Figure 27: La matrice de confusion pour comparer les valeurs de test prévues et réelles

Accuracy

```
Entrée [251]: ac = accuracy_score(y_test, y_pred)
              ac
Out[251]: 0.85
```

Figure 28:La précision

La précision est de 0,95.

Notre modèle fonctionne bien.

Conclusion :

Notre projet consiste à classer les publications sur les réseaux sociaux en utilisant l'algorithme KNN (K-nearest neighbor) avec le meilleur choix de k , pour cela le projet se divise de la sorte en 5 étapes.

La première étape constituant une description du dataset, on a fait dans cette étape l'importation des bibliothèques python nécessaires et la lecture des données à partir d'un fichier "Social_Network Ads.csv", on a affiché la dimension et les colonnes et les 5 premières lignes de notre dataset.

La deuxième étape intitulée Exploration de données, dans cette partie on a défini la présentation graphique de notre dataset.

Dans la troisième étape, on a construit notre modèle pour classer les publicités sur les réseaux sociaux.

Dans la quatrième étape, on a évalué notre modèle.

Dans la dernière étape on a fait un test de notre modèle.

En conclusion, ce fut une expérience très excitante et nous en avons beaucoup appris. Dans ce travail, nous avons découvert de nouveaux concepts qui nous aideront beaucoup dans le futur.

Références :

<https://fr.wikipedia.org/wiki>

<https://www.w3schools.com>

<https://www.geeksforgeeks.org>