université PARIS DIDEROT PARIS 7

CLASSIFICATION DES CATÉGORIES DE PRODUITS LORS D'UN BLACK-FRIDAY

Par HAMANI Khalil

Travail réalisé

- Classification de la catégorie d'un produit
  - *Modèles utilisés : SVM, Arbre de décision, forêt aléatoire*
- Régression sur le coût de la transaction
  - *Modèles utilisés : Régression linéaire, par Arbre de décision, par forêt aléatoire*

Présentation du dataset

- 1 ligne = 1 transaction

Présentation du dataset

- 1 ligne = 1 transaction
- ~ 537 k lignes

Présentation du dataset

- 1 ligne = 1 transaction
- ~ 537 k lignes
- Informations sur l'utilisateur : ID, Sexe, Age, Occupation (profession), État civil, Catégorie de la ville

Présentation du dataset

- 1 ligne = 1 transaction
- ~ 537 k lignes
- Informations sur l'utilisateur : ID, Sexe, Age, Occupation (profession), État civil, Catégorie de la ville
- Informations sur le produit : ID, Catégories (en trois colonnes ; Catégorie 1, Catégorie 2, Catégorie 3)

Présentation du dataset

- 1 ligne = 1 transaction
- ~ 537 k lignes
- Informations sur l'utilisateur : ID, Sexe, Age, Occupation (profession), État civil, Catégorie de la ville
- Informations sur le produit : ID, Catégories (en trois colonnes ; Catégorie 1, Catégorie 2, Catégorie 3)
- Coût de la transaction

Présentation du dataset

User_ID	Product_ID	Gender	Age	Occupation	City_Ca	Stay_	Marital_Stat	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
1000001	P00069042	F	0-17	10	A	2	0	3			8370
1000001	P00248942	F	0-17	10	A	2	0	1	6	14	15200
1000001	P00087842	F	0-17	10	A	2	0	12			1422
1000001	P00085442	F	0-17	10	A	2	0	12	14		1057
1000002	P00285442	M	55+	16	C	4+	0	8			7969
1000003	P00193542	M	26-35	15	A	3	0	1	2		15227
1000004	P00184942	M	46-50	7	B	2	1	1	8	17	19215
1000004	P00346142	M	46-50	7	B	2	1	1	15		15854
1000004	P0097242	M	46-50	7	B	2	1	1	16		15686
1000005	P00274942	M	26-35	20	A	1	1	8			7871
1000005	P00251242	M	26-35	20	A	1	1	5	11		5254
1000005	P00014542	M	26-35	20	A	1	1	8			3957
1000005	P00031342	M	26-35	20	A	1	1	8			6073
1000005	P00145042	M	26-35	20	A	1	1	1	2	5	15665
1000006	P00231342	F	51-55	9	A	1	0	5	8	14	5378
1000006	P00190242	F	51-55	9	A	1	0	4	5		2079
1000006	P0096642	F	51-55	9	A	1	0	2	3	4	13055
1000006	P00058442	F	51-55	9	A	1	0	5	14		8851
1000007	P00036842	M	36-45	1	B	1	1	1	14	16	11788
1000008	P00249542	M	26-35	12	C	4+	1	1	5	15	19614
1000008	P00220442	M	26-35	12	C	4+	1	5	14		8584
1000008	P00156442	M	26-35	12	C	4+	1	8			9872
1000008	P00213742	M	26-35	12	C	4+	1	8			9743

Présentation du dataset

Colonne	Type de données	Type de l'ensemble
User_ID	Entier	Discret
Product_ID	Entier	Discret
Gender (Sexe)	Chaine de caractères	Discret
Age (tranches d'ages)	Chaine de caractères	Discret
Occupation	Entier	Discret
City_Category	Caractère	Discret
Stay_In_Current_City_Years	Entier	Discret
Martial_Status	Entier	Discret
Product_Category_1	Entier	Discret
Product_Category_2	Entier	Discret
Product_Category_3	Entier	Discret
Purchase	Entier	Continu

Présentation du dataset

- Que pourrait-on faire avec ?

Présentation du dataset

- Que pourrait-on faire avec ?
  - *Classifications utiles : Age, Catégorie du produit*

Présentation du dataset

- Que pourrait-on faire avec ?
  - *Classifications utiles : Age, Catégorie du produit*
  - *Classifications inutiles : Sexe, Occupation (signification inconnue), État civil*

Présentation du dataset

- Que pourrait-on faire avec ?
  - *Classifications utiles : Age, Catégorie du produit*
  - *Classifications inutiles : Sexe, Occupation (signification inconnue), État civil*
  - *Régression : Coût de la transaction*

Présentation du dataset

- Que pourrait-on faire avec ?
  - *Classifications utiles : Age, Catégorie du produit*
  - *Classifications inutiles : Sexe, Occupation (signification inconnue), État civil*
  - *Régression : Coût de la transaction*
  - *Clusturing : Faire de la recommandation sur les produits*

Classification des catégories des produits

- Nettoyage du dataset (pré-traitements)
- Encodage
- Entrainement
- Mesures de performances

Outils utilisés

Classification de la catégorie

Entrées et sorties

User_ID	Product_ID	Gender	Age	Occupation	City_Cat	Stay_	Marital_Stat	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
1000001	P00069042	F	0-17	10	A	2	0	3			8370
1000001	P00248942	F	0-17	10	A	2	0	1	6	14	15200
1000001	P00087842	F	0-17	10	A	2	0	12			1422
1000001	P00085442	F	0-17	10	A	2	0	12	14		1057
1000002	P00285442	M	55+	16	C	4+	0	8			7969
1000003	P00193542	M	26-35	15	A	3	0	1	2		15227
1000004	P00184942	M	46-50	7	B	2	1	1	8	17	19215
1000004	P00346142	M	46-50	7	B	2	1	1	15		15854
1000004	P0097242	M	46-50	7	B	2	1	1	16		15686
1000005	P00274942	M	26-35	20	A	1	1	8			7871
1000005	P00251242	M	26-35	20	A	1	1	5	11		5254
1000005	P00014542	M	26-35	20	A	1	1	8			3957
1000005	P00031342	M	26-35	20	A	1	1	8			6073
1000005	P00145042	M	26-35	20	A	1	1	1	2	5	15665
1000006	P00231342	F	51-55	9	A	1	0	5	8	14	5378
1000006	P00190242	F	51-55	9	A	1	0	4	5		2079
1000006	P0096642	F	51-55	9	A	1	0	2	3	4	13055
1000006	P00058442	F	51-55	9	A	1	0	5	14		8851
1000007	P00036842	M	36-45	1	B	1	1	1	14	16	11788
1000008	P00249542	M	26-35	12	C	4+	1	1	5	15	19614
1000008	P00220442	M	26-35	12	C	4+	1	5	14		8584
1000008	P00156442	M	26-35	12	C	4+	1	8			9872
1000008	P00213742	M	26-35	12	C	4+	1	8			9743

Entrées

Entrées et sorties

Sortie

User_ID	Product_ID	Gender	Age	Occupation	City_Cat	Stay_	Marital_Stat	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
1000001	P00069042	F	0-17	10	A	2	0	3			8370
1000001	P00248942	F	0-17	10	A	2	0	1	6	14	15200
1000001	P00087842	F	0-17	10	A	2	0	12			1422
1000001	P00085442	F	0-17	10	A	2	0	12	14		1057
1000002	P00285442	M	55+	16	C	4+	0	8			7969
1000003	P00193542	M	26-35	15	A	3	0	1	2		15227
1000004	P00184942	M	46-50	7	B	2	1	1	8	17	19215
1000004	P00346142	M	46-50	7	B	2	1	1	15		15854
1000004	P0097242	M	46-50	7	B	2	1	1	16		15686
1000005	P00274942	M	26-35	20	A	1	1	8			7871
1000005	P00251242	M	26-35	20	A	1	1	5	11		5254
1000005	P00014542	M	26-35	20	A	1	1	8			3957
1000005	P00031342	M	26-35	20	A	1	1	8			6073
1000005	P00145042	M	26-35	20	A	1	1	1	2	5	15665
1000006	P00231342	F	51-55	9	A	1	0	5	8	14	5378
1000006	P00190242	F	51-55	9	A	1	0	4	5		2079
1000006	P0096642	F	51-55	9	A	1	0	2	3	4	13055
1000006	P00058442	F	51-55	9	A	1	0	5	14		8851
1000007	P00036842	M	36-45	1	B	1	1	1	14	16	11788
1000008	P00249542	M	26-35	12	C	4+	1	1	5	15	19614
1000008	P00220442	M	26-35	12	C	4+	1	5	14		8584
1000008	P00156442	M	26-35	12	C	4+	1	8			9872
1000008	P00213742	M	26-35	12	C	4+	1	8			9743

Entrées

Pré-traitements (Catégorie)

```
In [3]: bf_df = pd.read_csv("./BlackFriday.csv")
        bf_df.head()
```

Out[3]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN	
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN	
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	
4	1000002	P00285442	M	55+	16	C	4+	0	8	NaN	

- Remplissage des vides par des 0

```
bf_df.fillna(0).head()
```

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_
0	1000001	P00069042	F	0-17	10	A	2	0	3	0.0	
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	
2	1000001	P00087842	F	0-17	10	A	2	0	12	0.0	
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	
4	1000002	P00285442	M	55+	16	C	4+	0	8	0.0	

Pré-traitements (Catégorie)

- Encodage des catégories
  - *Catégorie 1 = {1, …, 18}* => *Catégorie 1 = {A, …, R}*
  - *Catégorie 2 = {2, …, 18}* => *Catégorie 1 = {B, …, R, 0}*
  - *Catégorie 3 = {3, …, 18}* => *Catégorie 1 = {C, …, R, 0}*

Pré-traitements (Catégorie)

- Encodage des catégories
  - *Catégorie 1 = {1, …, 18}* => *Catégorie 1 = {A, …, R}*
  - *Catégorie 2 = {2, …, 18}* => *Catégorie 1 = {B, …, R, 0}*
  - *Catégorie 3 = {3, …, 18}* => *Catégorie 1 = {C, …, R, 0}*
- Concaténation des catégories
  - *Product_Category : Catégorie 1 + Catégorie 2 + Catégorie 3*
    - Exemple :
      - *Catégorie 1 : 1 => A*
      - *Catégorie 2 : 2 => B* => *Product_Category : ABC*
      - *Catégorie 3 : 3 => C*

*Toutes les combinaisons donnent 5508 catégories

Pré-traitements (Catégorie)

- Encodage des catégories
  - *Catégorie 1 = {1, …, 18}* => *Catégorie 1 = {A, …, R}*
  - *Catégorie 2 = {2, …, 18}* => *Catégorie 1 = {B, …, R, 0}*
  - *Catégorie 3 = {3, …, 18}* => *Catégorie 1 = {C, …, R, 0}*
- Concaténation des catégories
  - *Product_Category : Catégorie 1 + Catégorie 2 + Catégorie 3*
    - Exemple :
      - *Catégorie 1 : 1 => A*
      - *Catégorie 2 : 2 => B* => *Product_Category : ABC*
      - *Catégorie 3 : 3 => C*

*Toutes les combinaisons donnent 5508 catégories **C'est beaucoup!!!**

Pré-traitements (Catégorie)

- Encodage des catégories
  - *Catégorie 1 = {1, …, 18}* => *Catégorie 1 = {A, …, R}*
  - *Catégorie 2 = {2, …, 18}* => *Catégorie 1 = {B, …, R, 0}*
  - *Catégorie 3 = {3, …, 18}* => *Catégorie 1 = {C, …, R, 0}*
- Concaténation des catégories
  - *Product_Category : Catégorie 1 + Catégorie 2 + Catégorie 3*
    - Exemple :
      - *Catégorie 1 : 1 => A*
      - *Catégorie 2 : 2 => B* => *Product_Category : ABC*
      - *Catégorie 3 : 3 => C*

*Toutes les combinaisons donnent 5508 catégories **C'est beaucoup!!!**

*Dans le dataset, il y'a 235 catégories

Pré-traitements (Catégorie)

- Encodage des catégories
  - *Catégorie 1 = {1, ..., 18}* => *Catégorie 1 = {A, ..., R}*
  - *Catégorie 2 = {2, ..., 18}* => *Catégorie 1 = {B, ..., R, 0}*
  - *Catégorie 3 = {3, ..., 18}* => *Catégorie 1 = {C, ..., R, 0}*
- Concaténation des catégories
  - *Product_Category : Catégorie 1 + Catégorie 2 + Catégorie 3*
    - Exemple :
      - *Catégorie 1 : 1 => A*
      - *Catégorie 2 : 2 => B* => *Product_Category : ABC*
      - *Catégorie 3 : 3 => C*

*Toutes les combinaisons donnent 5508 catégories **C'est beaucoup!!!**

*Dans le dataset, il y'a 235 catégories **C'est beaucoup aussi... Mais... Moins beaucoup qu'avant quand même :D**

Pré-traitements (Catégorie)

- Suppression des doublons par combinaison
  - *Exemple : Les catégories suivantes sont considérées comme étant la même catégorie*
    - DEF, EDF, EFD, DFE, FDE, FED
- Après ce pré-traitement le nombre de catégories est resté le même i.e. 235

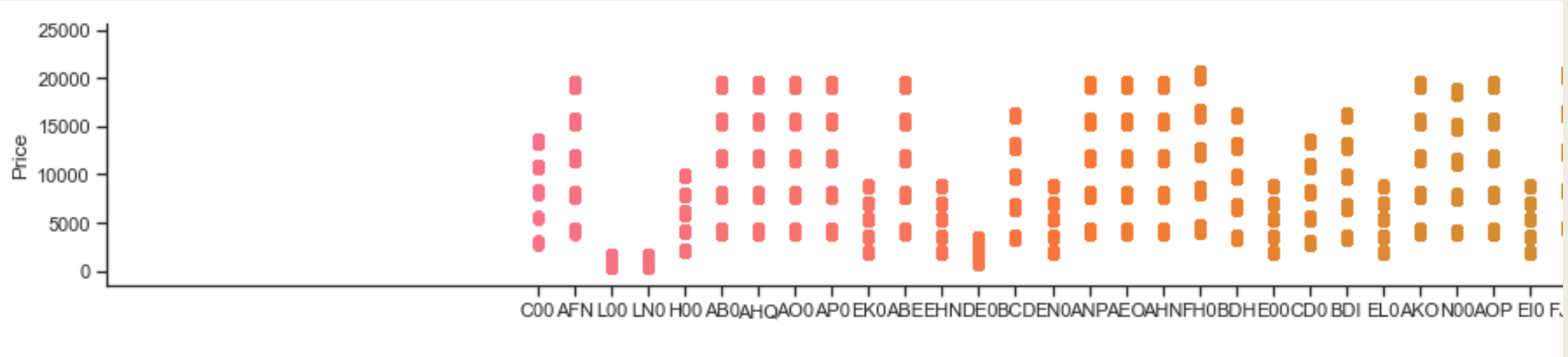
Pré-traitements (User_ID)

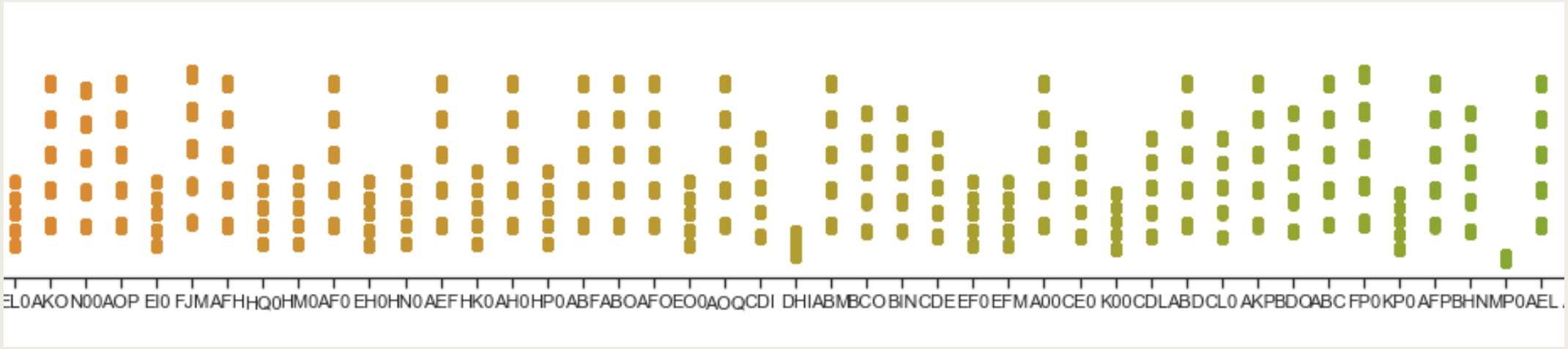
- Tout simplement supprimé pour cause d’impretinence

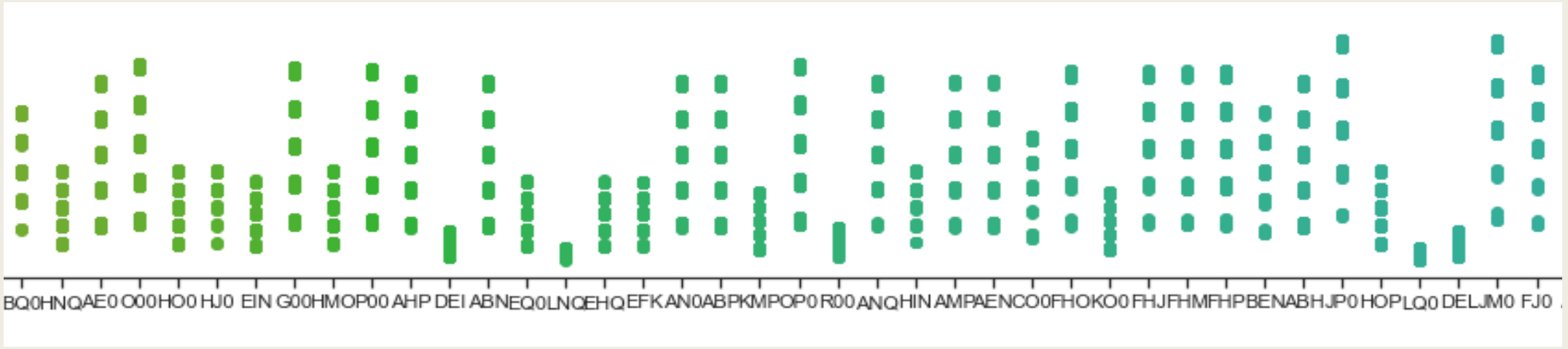
Encodage

- Séquentiel et par ordre d'apparition : 0, 1, 2, 3… N pour toutes les colonnes à l'exception du coût de la transaction

- Le coût de la transaction a été encodé en utilisant un normalisation par Minimum (=0) et Maximum (=1)




EL0 AKO N00 AOP EI0 FJM AFH HQ0 HM0 AF0 EH0 HN0 AEF HK0 AH0 HP0 ABF ABO AFO EO0 AOQ CDI DHI ABM BCO BIN CDE EF0 EFM A00 CE0 K00 CDL ABD CL0 AKP BDO ABC FP0 KP0 AFP BHN MP0 AEL


BQ0HNQAE0 O00HO0 HJ0 EIN G00HMOP00 AHP DEI ABNEQ0LNQEHQ EFK AN0ABPKMPOP0 R00 ANQ HIN AMPAENCO0FHOKO0 FHJFHMFHP BENAB HJP0 HOP LQ0 DELJM0 FJ0

Division du dataset

- Division du dataset en 70% train et 30% test avec l'option « shuffle » pour prendre au hasard et non de manière séquentielle
- Pour cette étape la fonction « train_test_split » de la bibliothèque « sklearn.model_selection » a été utilisée parce qu'elle découpe en gardant une homogénéité dans les deux ensembles par rapport à l'ensemble source

Classification (SVM)

- Précision moyenne : 0.2%
- Recall moyen : 0.6%

Classification (SVM)

- Précision moyenne : 0.2%
- Recall moyen : 0.6%

Classification (Arbre de décision)

- Accuracy : 97% (Is this overfiting ?!!!)
- Précision moyenne : 93.7%
- Recall moyen : 93.5%

Classification (Arbre de décision)

Profondeur plafonnée à 21

- Accuracy : 85%
- Précision moyenne : 87.1%
- Recall moyen : 73%

Classification (Forêt aléatoire)

Profondeur plafonnée à 25

Avec 63 arbres

- Accuracy : 45%
- Précision moyenne : 29.2%
- Recall moyen : 20.46%

```
1. in precision :  0
0. in precision :  40
1. in recall :  0
0. in recall :  40
```

Classification (Forêt aléatoire)

- Accuracy : 42%
- Précision moyenne : 25.4%
- Recall moyen : 19.7%

Regression sur le coût d’une transaction

- Pré-traitements
- Encodage
- Entrainement
- Mesures de performances

Entrées et sorties

User_ID	Product_ID	Gender	Age	Occupation	City_Cat	Stay_	Marital_Stati	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
1000001	P00069042	F	0-17	10	A	2	0	3			8370
1000001	P00248942	F	0-17	10	A	2	0	1	6	14	15200
1000001	P00087842	F	0-17	10	A	2	0	12			1422
1000001	P00085442	F	0-17	10	A	2	0	12	14		1057
1000002	P00285442	M	55+	16	C	4+	0	8			7969
1000003	P00193542	M	26-35	15	A	3	0	1	2		15227
1000004	P00184942	M	46-50	7	B	2	1	1	8	17	19215
1000004	P00346142	M	46-50	7	B	2	1	1	15		15854
1000004	P0097242	M	46-50	7	B	2	1	1	16		15686
1000005	P00274942	M	26-35	20	A	1	1	8			7871
1000005	P00251242	M	26-35	20	A	1	1	5	11		5254
1000005	P00014542	M	26-35	20	A	1	1	8			3957
1000005	P00031342	M	26-35	20	A	1	1	8			6073
1000005	P00145042	M	26-35	20	A	1	1	1	2	5	15665
1000006	P00231342	F	51-55	9	A	1	0	5	8	14	5378
1000006	P00190242	F	51-55	9	A	1	0	4	5		2079
1000006	P0096642	F	51-55	9	A	1	0	2	3	4	13055
1000006	P00058442	F	51-55	9	A	1	0	5	14		8851
1000007	P00036842	M	36-45	1	B	1	1	1	14	16	11788
1000008	P00249542	M	26-35	12	C	4+	1	1	5	15	19614
1000008	P00220442	M	26-35	12	C	4+	1	5	14		8584
1000008	P00156442	M	26-35	12	C	4+	1	8			9872
1000008	P00213742	M	26-35	12	C	4+	1	8			9743

Entrées

Entrées et sorties

Sortie

User_ID	Product_ID	Gender	Age	Occupation	City_Cat	Stay_	Marital_Stat	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
1000001	P00069042	F	0-17	10	A	2	0	3			8370
1000001	P00248942	F	0-17	10	A	2	0	1	6	14	15200
1000001	P00087842	F	0-17	10	A	2	0	12			1422
1000001	P00085442	F	0-17	10	A	2	0	12	14		1057
1000002	P00285442	M	55+	16	C	4+	0	8			7969
1000003	P00193542	M	26-35	15	A	3	0	1	2		15227
1000004	P00184942	M	46-50	7	B	2	1	1	8	17	19215
1000004	P00346142	M	46-50	7	B	2	1	1	15		15854
1000004	P0097242	M	46-50	7	B	2	1	1	16		15686
1000005	P00274942	M	26-35	20	A	1	1	8			7871
1000005	P00251242	M	26-35	20	A	1	1	5	11		5254
1000005	P00014542	M	26-35	20	A	1	1	8			3957
1000005	P00031342	M	26-35	20	A	1	1	8			6073
1000005	P00145042	M	26-35	20	A	1	1	1	2	5	15665
1000006	P00231342	F	51-55	9	A	1	0	5	8	14	5378
1000006	P00190242	F	51-55	9	A	1	0	4	5		2079
1000006	P0096642	F	51-55	9	A	1	0	2	3	4	13055
1000006	P00058442	F	51-55	9	A	1	0	5	14		8851
1000007	P00036842	M	36-45	1	B	1	1	1	14	16	11788
1000008	P00249542	M	26-35	12	C	4+	1	1	5	15	19614
1000008	P00220442	M	26-35	12	C	4+	1	5	14		8584
1000008	P00156442	M	26-35	12	C	4+	1	8			9872
1000008	P00213742	M	26-35	12	C	4+	1	8			9743

Entrées

Pré-traitements

Lecture du DataSet

In [30]:
```
bf_df = pd.read_csv("./BlackFriday.csv")
bf_df = bf_df.fillna(0)
bf_df.head()
```

Out[30]:

_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
)42	F	0-17	10	A	2	0	3	0.0	0.0	8370
)42	F	0-17	10	A	2	0	1	6.0	14.0	15200
)42	F	0-17	10	A	2	0	12	0.0	0.0	1422
)42	F	0-17	10	A	2	0	12	14.0	0.0	1057
)42	M	55+	16	C	4+	0	8	0.0	0.0	7969

In [31]:
```
bf_df["Product_Category_2"] = bf_df["Product_Category_2"].astype(int)
bf_df["Product_Category_3"] = bf_df["Product_Category_3"].astype(int)
bf_df = bf_df.drop(columns=["User_ID"])
bf_df.head()
```

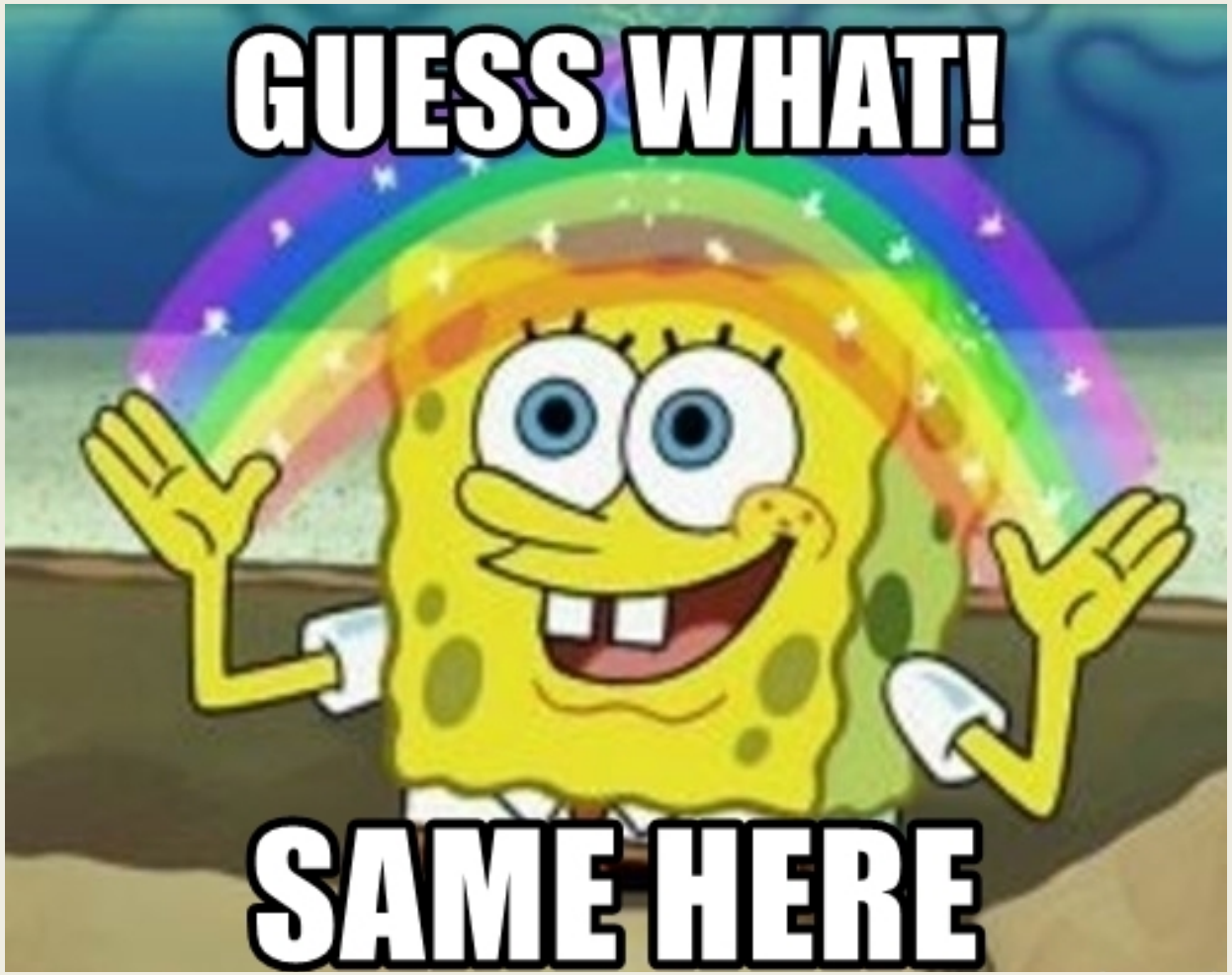
Out[31]:

_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
)42	F	0-17	10	A	2	0	3	0	0	8370
)42	F	0-17	10	A	2	0	1	6	14	15200
)42	F	0-17	10	A	2	0	12	0	0	1422
)42	F	0-17	10	A	2	0	12	14	0	1057
)42	M	55+	16	C	4+	0	8	0	0	7969

In [32]:
```
bf_df.to_csv("./BlackFriday_Price_Preprocessed.csv", index=False)
```

Encodage

Encodage



Régression linéaire

- Score : 0.6%

Linear Regression

Entrainement

```
In [16]: reg = LinearRegression()

         reg = train_model(reg, X_train, y_train)
```

```
Model entrainé en : 00:00:00
```

Mesures de performances

```
In [17]: reg.score(X_test, y_test)
```

```
Out[17]: 0.059940413938787 46
```

Régression par arbre de décision

- Pas de limitation sur la profondeur de l'arbre :
  - *Score : 43%*

- Après limitation de la profondeur à 13 :
  - *Score : 68%*

Régression par forêt aléatoire

- Pas de limitation sur la profondeur de l'arbre :
  - *Score : 43%*
- Après limitation de la profondeur à 13 :
  - *Score : 70%*

***Nombre d'arbres = 200**

Conclusions et perspectives

- SVM et la régression linéaire ont donnés de mauvais résultats… Pourquoi ?!

Conclusions et perspectives

- SVM et la régression linéaire ont donnés de mauvais résultats… Pourquoi ?!
- Continuer à chercher les bons paramètres des modèles pour de meilleurs résultats

Conclusions et perspectives

- SVM et la régression linéaire ont donnés de mauvais résultats… Pourquoi ?!
- Continuer à chercher les bons paramètres des modèles pour de meilleurs résultats
- Certaines catégories n’ont pas beaucoup d’instances

Conclusions et perspectives

- SVM et la régression linéaire ont donnés de mauvais résultats… Pourquoi ?!
- Continuer à chercher les bons paramètres des modèles pour de meilleurs résultats
- Certaines catégories n’ont pas beaucoup d’instances
- Faire une récolte pour avoir plus de données

Conclusions et perspectives

- SVM et la régression linéaire ont donnés de mauvais résultats… Pourquoi ?!
- Continuer à chercher les bons paramètres des modèles pour de meilleurs résultats
- Certaines catégories n’ont pas beaucoup d’instances
- Faire une récolte pour avoir plus de données
- Essayer une validation croisée peut-être ?