Projet Data Mining

Janvier 17, 2024

Préparé par:
Abdelkader Fourati & Oussema Makhlouf
2ème année cycle d'ingénieur

Intro

-Dans le cadre de ce projet, nous avons entrepris l'analyse, la visualisation et le Prétraitement des données fournies par une compagnie d'assurance. L'objectif était de développer un modèle performant capable de prédire si un bâtiment aura un accident pendant la période d’assurance.

-Nous avons utilisé les bibliothèques numpy, pandas, matplotlib, seaborn et scikit-learn (sklearn) de Python pour atteindre cet objectif. De plus, nous avons exploité Google Colab comme environnement de développement collaboratif en ligne.

```
path1="/content/drive/MyDrive/train_Insurance.csv"
df_train=pd.read_csv(path1)
df_train.head()
```

	Customer Id	YearOfObservation	Insured_Period	Residential	Building_Painted	Building_Fenced	Garden	Settlement	Building Dimension	Building_Type	NumberOfWindows	Geo_Code	Claim
0	H13501	2012	1.0	1	N	V	V	U	1240.0	Wood-framed	without	75117	non
1	H14962	2012	1.0	0	N	V	V	U	900.0	Non-combustible	without	62916	non
2	H17755	2013	1.0	1	V	N	O	R	4984.0	Non-combustible	4	31149	oui
3	H13369	2016	0.5	0	N	V	V	U	600.0	Wood-framed	without	6012	oui
4	H12988	2012	1.0	0	N	V	V	U	900.0	Non-combustible	without	57631	non

```
[ ] #nbre d'observation
    print(len(df_train))
```

```
5012
```

Le jeu de données d'entraînement (train) est constitué de 12 attributs descripteurs et d'une variable classe 'claim'. Il comporte également 5012 observations.

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5012 entries, 0 to 5011
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Customer Id         5012 non-null   object 
 1   YearOfObservation   5012 non-null   int64  
 2   Insured_Period      5012 non-null   float64
 3   Residential         5012 non-null   int64  
 4   Building_Painted    5012 non-null   object 
 5   Building_Fenced     5012 non-null   object 
 6   Garden              5008 non-null   object 
 7   Settlement          5012 non-null   object 
 8   Building Dimension  4935 non-null   float64
 9   Building_Type       5012 non-null   object 
 10  NumberOfWindows     5012 non-null   object 
 11  Geo_Code            4939 non-null   object 
 12  Claim               5012 non-null   object 
dtypes: float64(2), int64(2), object(9)
memory usage: 509.2+ KB
```

Parmi les 13 attributs, 2 sont de type float, 2 de type int, et 9 de type object.

```
df_train.describe(include="all")
```

	Customer Id	YearOfObservation	Insured_Period	Residential	Building_Painted	Building_Fenced	Garden	Settlement	Building Dimension	Building_Type	NumberOfWindows	Geo_Code	Claim
count	5012	5012.000000	5012.000000	5012.000000	5012	5012	5008	5012	4935.000000	5012	5012	4939	5012
unique	5012	NaN	NaN	NaN	2	2	2	2	NaN	4	11	1115	2
top	H13501	NaN	NaN	NaN	V	N	O	R	NaN	Non-combustible	without	6088	non
freq	1	NaN	NaN	NaN	3763	2535	2532	2537	NaN	2310	2476	102	3886
mean	NaN	2013.660215	0.869713	0.301077	NaN	NaN	NaN	NaN	1876.898683	NaN	NaN	NaN	NaN
std	NaN	1.383134	0.219496	0.458772	NaN	NaN	NaN	NaN	2267.277397	NaN	NaN	NaN	NaN
min	NaN	2012.000000	0.500000	0.000000	NaN	NaN	NaN	NaN	1.000000	NaN	NaN	NaN	NaN
25%	NaN	2012.000000	0.500000	0.000000	NaN	NaN	NaN	NaN	520.000000	NaN	NaN	NaN	NaN
50%	NaN	2013.000000	1.000000	0.000000	NaN	NaN	NaN	NaN	1067.000000	NaN	NaN	NaN	NaN
75%	NaN	2015.000000	1.000000	1.000000	NaN	NaN	NaN	NaN	2280.000000	NaN	NaN	NaN	NaN
max	NaN	2016.000000	1.000000	1.000000	NaN	NaN	NaN	NaN	20840.000000	NaN	NaN	NaN	NaN

-Les attributs ‘Building_Painted’,’Building_Fenced’,’Garden’,’Settlement’ et ‘Claim’ possèdent 2 éléments uniques, ‘Building_Type’ possède 4 élements.
-la valeur la plus fréquente de ‘Building_Type’ est ‘non-combustible’,on constate aussi qu’il se trouve dans presque la moitié des cas.
-la valeur la plus fréquente de ‘NumberOfWindows’ est ‘without’,on constate aussi qu’il se trouve dans presque la moitié des cas.

	Customer Id	YearOfObservation	Insured_Period	Residential	Building_Painted	Building_Fenced	Garden	Settlement	Building Dimension	Building_Type	NumberOfWindows	Geo_Code	Claim
count	5012	5012.000000	5012.000000	5012.000000	5012	5012	5008	5012	4935.000000	5012	5012	4939	5012
unique	5012	NaN	NaN	NaN	2	2	2	2	NaN	4	11	1115	2
top	H13501	NaN	NaN	NaN	V	N	O	R	NaN	Non-combustible	without	6088	non
freq	1	NaN	NaN	NaN	3763	2535	2532	2537	NaN	2310	2476	102	3886
mean	NaN	2013.660215	0.869713	0.301077	NaN	NaN	NaN	NaN	1876.898683	NaN	NaN	NaN	NaN

-Une constatation très importante içi, la valeur la plus fréquente de la variable classe ‘Claim’ qui est ‘non’, présente plus de la moitié des observations.
=>donc un déséquilibre entre les classes ‘oui’ et ‘non’.Par conséquent, on se contente pas seulement de la valeur de l’accuracy dans l’évaluation du modèle.

```
df_train.isna().sum()
```

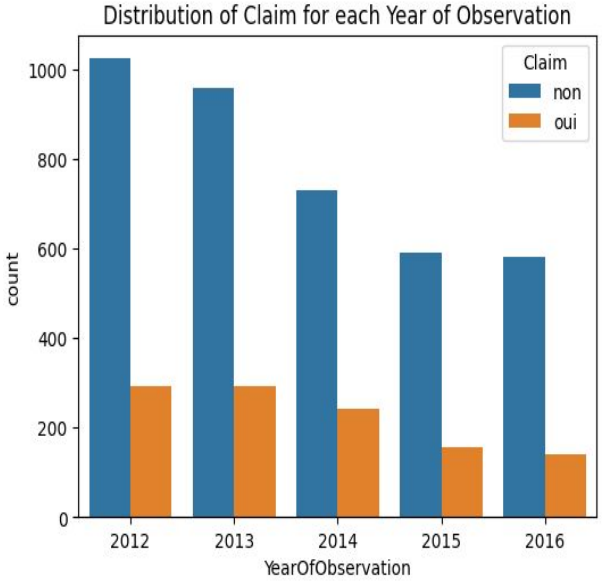
```
Customer Id           0
YearOfObservation     0
Insured_Period        0
Residential           0
Building_Painted      0
Building_Fenced       0
Garden                4
Settlement            0
Building Dimension    77
Building_Type         0
NumberOfWindows       0
Geo_Code              73
Claim                 0
dtype: int64
```

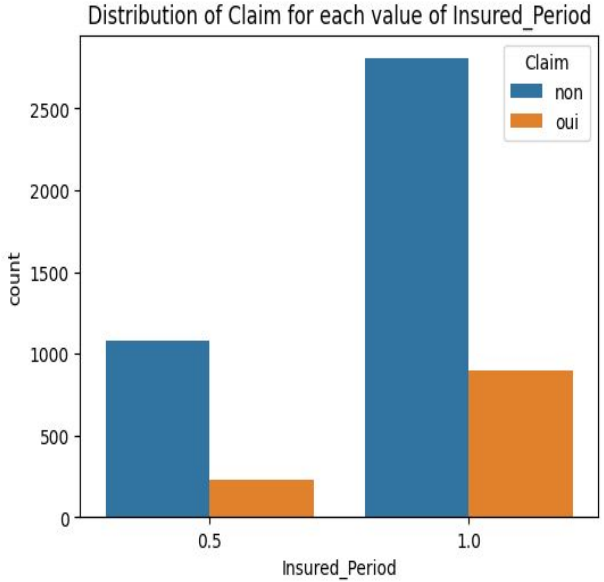
```
df_train.isna().sum().sum()
```

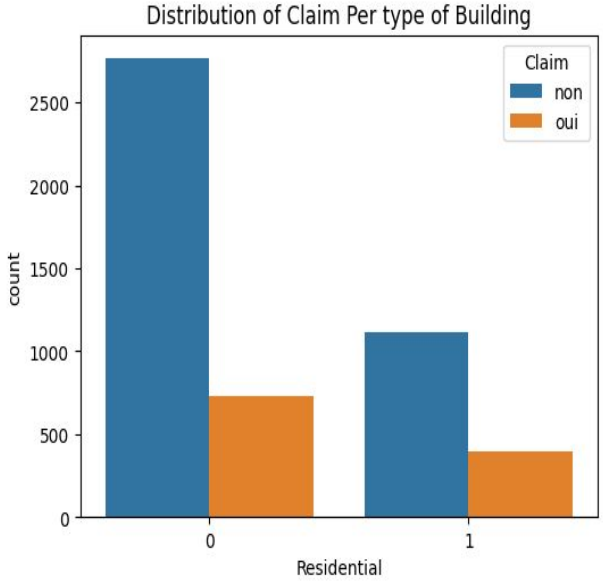
```
154
```


Certains attributs présentent des valeurs manquantes (NAN), il est donc nécessaire de remplacer ces NAN par des valeurs bien définies en utilisant les techniques appropriées.

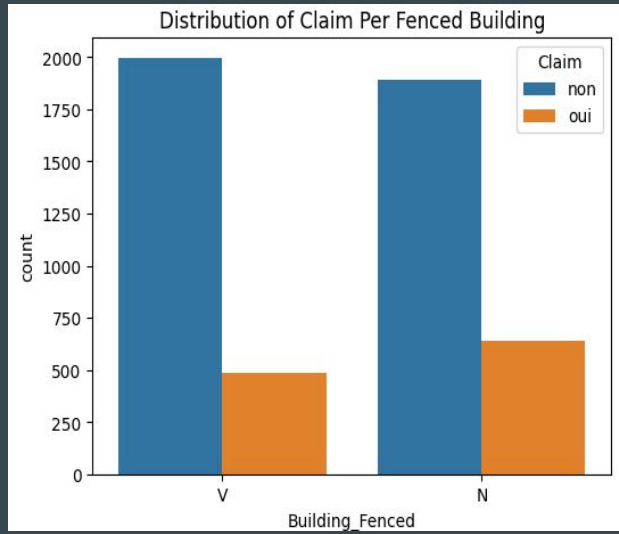
Data Visualisation

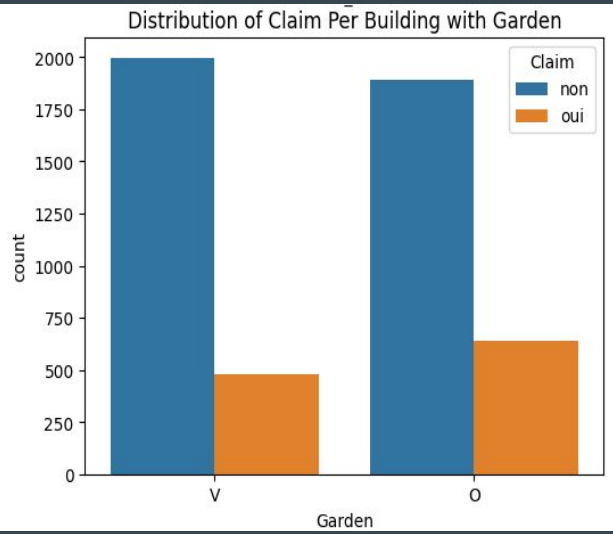


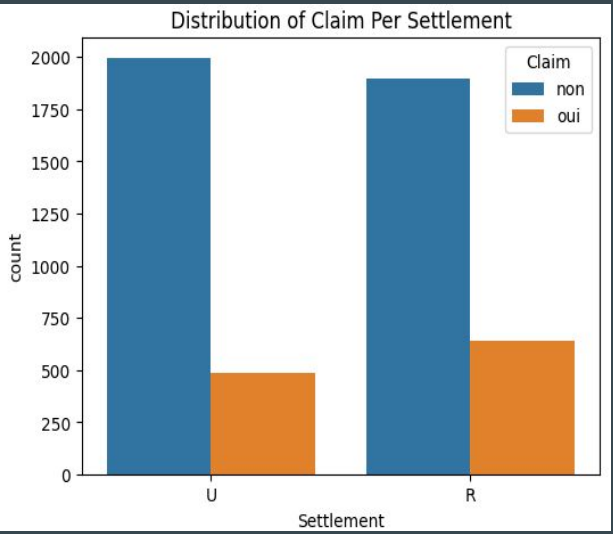




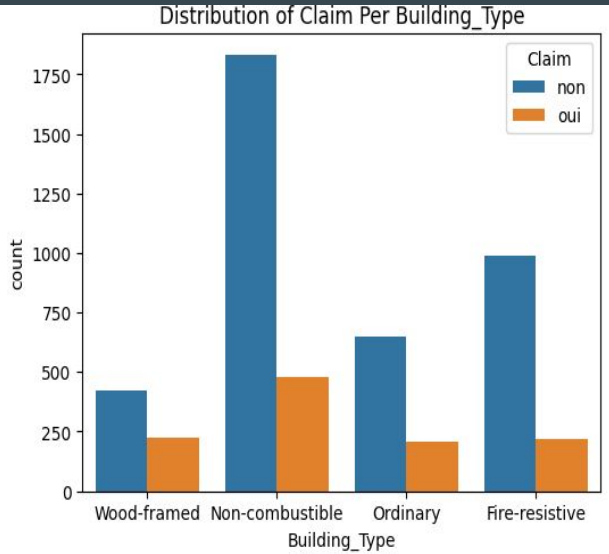
-le 1èr countplot montre que 2012 était l'année dont l'assurance a fait plus d'observations d'etat de bâtiment, et chaque année le nombre d'observations diminue pour rester stable pour les 2 années 2015 et 2016.
-le 2eme countplot montre que l'assurance accorde plus de contrat à 1an que à 6 mois et que 15% des périodes de 6 mois ont une ou plusieurs réclamations,par contre 25% des périodes d'1 an ont une ou plusieurs réclamations.
-le 3ème countplot montre que l'assurance accorde plus de contrats aux bâtiments non résidentiels puisqu'ils présentent moins de réclamations.

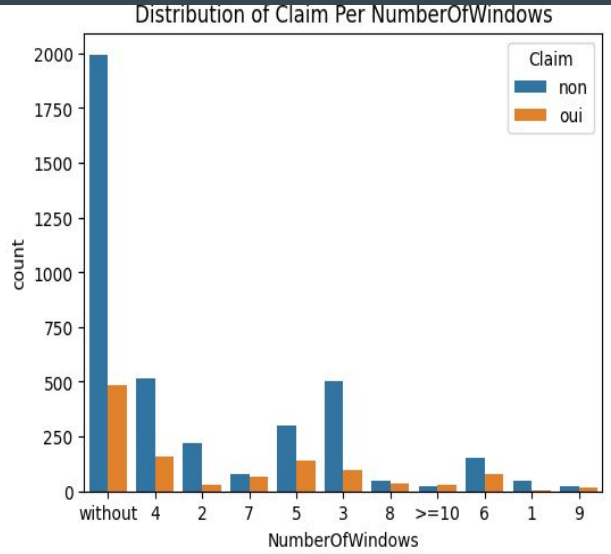






-les bâtiments sans jardin présentent plus de réclamations.
-Les bâtiments dans les zones rurales présentent plus de réclamations.





-l'assurance accorde plus de contrat aux bâtiments de type 'non-combustible' et qui ne comportent pas de fenêtres.

Data preprocessing

```
df_train.isna().sum()
```

```
Customer Id           0
YearOfObservation     0
Insured_Period        0
Residential           0
Building_Painted      0
Building_Fenced       0
Garden                4
Settlement            0
Building Dimension    77
Building_Type         0
NumberOfWindows       0
Geo_Code              73
Claim                 0
dtype: int64
```

```
Customer Id           0
YearOfObservation     0
Insured_Period        0
Residential           0
Building_Painted      0
Building_Fenced       0
Garden                0
Settlement            0
Building Dimension    0
Building_Type         0
NumberOfWindows       0
Geo_Code              0
Claim                 0
dtype: int64
```

-On a remplacé les NAN dans ‘Garden’ par la valeur la plus fréquente de cet attribut, dans ‘Building Dimension’ par la valeur médiane et dans ‘Geo_code’ par 0.

	YearOfObservation	Insured_Period	Residential	Building_Painted	Building_Fenced	Garden	Settlement	Building Dimension	Building_Type	NumberOfWindows	Claim
0	2012	1.0	1	0.0	1.0	1.0	0.0	2.0	3.0	0	non
1	2012	1.0	0	0.0	1.0	1.0	0.0	1.0	1.0	0	non
2	2013	1.0	1	1.0	0.0	0.0	1.0	3.0	1.0	4	oui
3	2016	0.5	0	0.0	1.0	1.0	0.0	1.0	3.0	0	oui
4	2012	1.0	0	0.0	1.0	1.0	0.0	1.0	1.0	0	non
...	...	...	...	...	...	...	...	...	...	...	...
5007	2013	1.0	0	0.0	1.0	1.0	0.0	1.0	2.0	0	oui
5008	2012	0.5	0	1.0	0.0	0.0	1.0	1.0	0.0	4	non
5009	2015	1.0	1	1.0	0.0	0.0	1.0	0.0	2.0	3	non
5010	2012	0.5	0	1.0	0.0	0.0	1.0	1.0	0.0	4	non
5011	2013	1.0	1	1.0	1.0	1.0	0.0	2.0	3.0	0	non

5012 rows × 11 columns

-On a encodé les attributs catégorielles en utilisant l'ordinal encoder, et on a spécifier l'ordre aussi.
-On a discrétisé les valeurs de l'attribut 'Building Dimension' en utilisant KBinsDiscretizer et la stratégie 'quantile' puisque la distribution est asymétrique.
-On a remplacé "without" dans l'attribut "NumberOfWindows" par 0 et on a rendu cet attribut de type numérique.
-On a supprimé les attributs "Customer Id" et "Geo Code".

```
thresholder = VarianceThreshold()
v = thresholder.fit(df_train.iloc[:,:-1])
print(v.variances_)
```

```
[1.91267896 0.04816887 0.2104298  0.18710032 0.24996652 0.24996417
 0.24996174 1.25026513 0.88918069 6.42793035]
```

Les descripteurs les plus discriminants sont: “YearOfObservation”, “Building Dimension”,”NumberOfWindows”.

```
df_test
```

	YearOfObservation	Insured_Period	Residential	Building_Painted	Building_Fenced	Garden	Settlement	Building Dimension	Building_Type	NumberOfWindows	Claim
0	2013	1.0	0	1.0	1.0	1.0	0.0	3.0	0.0	0	oui
1	2015	1.0	0	1.0	0.0	0.0	1.0	2.0	0.0	5	non
2	2013	1.0	1	1.0	0.0	0.0	1.0	2.0	2.0	6	oui
3	2015	1.0	0	0.0	1.0	1.0	0.0	2.0	1.0	0	oui
4	2016	0.5	0	1.0	0.0	0.0	1.0	3.0	0.0	9	non
...	...	...	...	...	...	...	...	...	...	...	...
2142	2016	0.5	1	1.0	0.0	0.0	1.0	1.0	3.0	2	non
2143	2012	1.0	0	1.0	1.0	1.0	0.0	2.0	1.0	0	non
2144	2014	1.0	0	1.0	0.0	0.0	1.0	1.0	1.0	3	non
2145	2014	1.0	1	0.0	1.0	1.0	0.0	1.0	1.0	0	oui
2146	2013	0.5	0	0.0	1.0	1.0	0.0	1.0	1.0	0	non

2147 rows × 11 columns

-On a effectué les même opérations sur le data du test.

Model Training

```
cls = DecisionTreeClassifier()
cls.fit(X_train, Y_train)
```

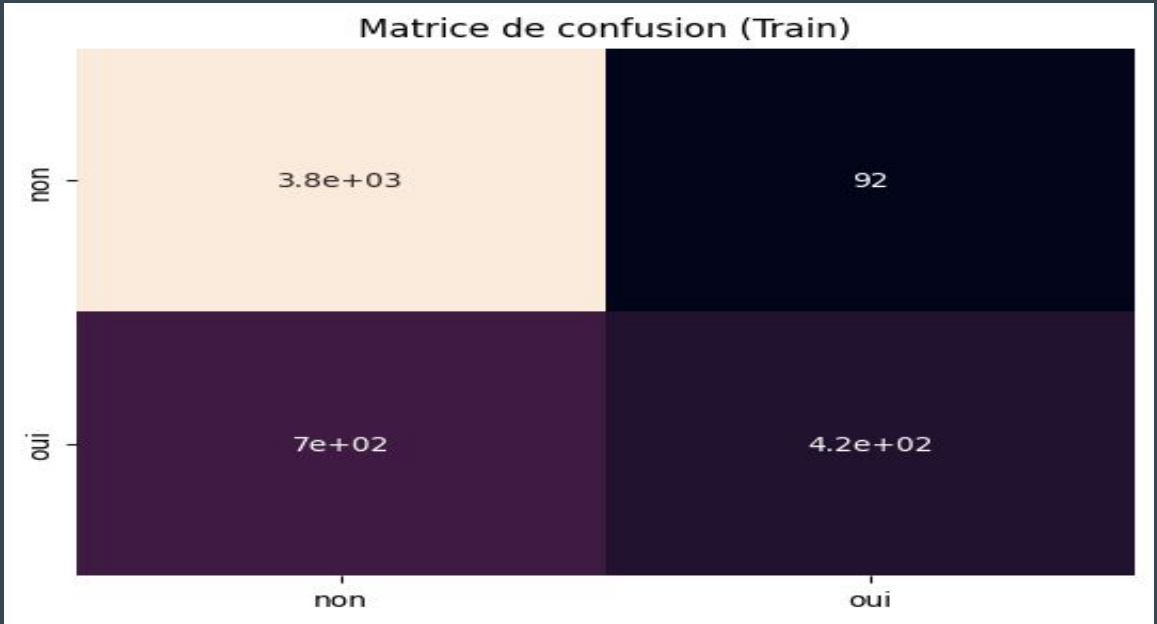
```
DecisionTreeClassifier()
```

+ Code + Text

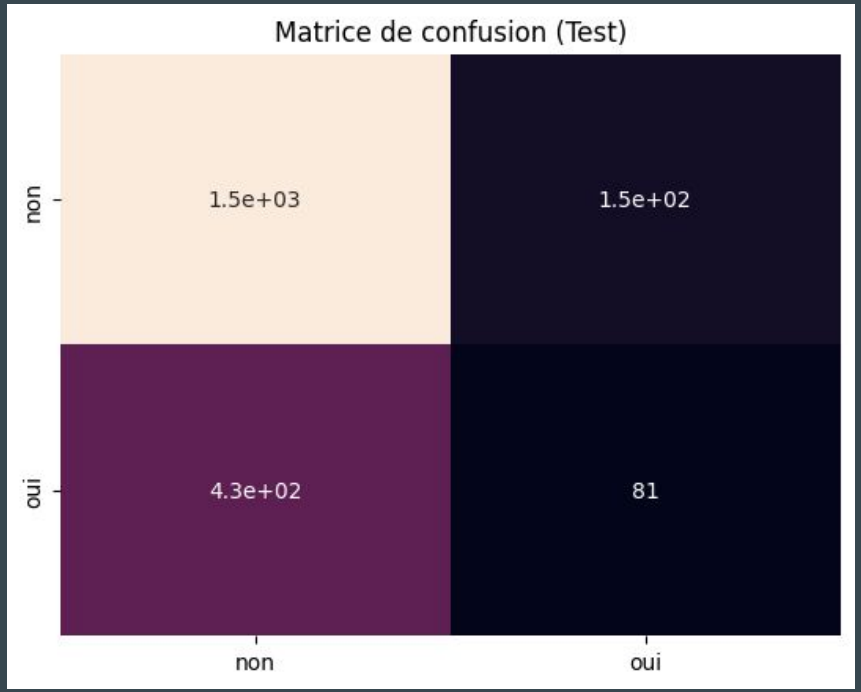
```
print(cls.classes_)
print(list(df_train))
```

```
['non' 'oui']
['YearOfObservation', 'Insured_Period', 'Residential', 'Building_Painted', 'Building_Fenced', 'Garden', 'Settlement', 'Building Dimension', 'Building_Type', 'NumberOfWindows', 'Claim'
```


-On a appliqué DecisionTreeClassifier comme technique d'apprentissage supervisée pour générer un modèle de prédiction.



La matrice de confusion du Train montre que:
-le modèle a peu de False positif (FP).
-Le modèle semble avoir une sensibilité relativement faible aux faux négatifs, ce qui signifie qu'il a du mal à détecter certaines réclamations réelles. Cela pourrait être particulièrement préoccupant en fonction des coûts associés aux réclamations non détectées.



La matrice de confusion du Test montre que:
-le modèle a mal à détecter les True Positive(TP).
-la même remarque pour les False Négatif que celle du data du train.
-beaucoup de False Positif ce qui va donner une mauvaise impression du bâtiment pour l'assurance.

```
[ ] from sklearn.metrics import precision_recall_fscore_support, roc_curve, roc_auc_score

    precision,recall,f1,occ= precision_recall_fscore_support(Y_test,Y_p_test, average=None)
    print(precision, recall, f1, occ)
```

```
[0.77713987 0.35064935] [0.90848078 0.15944882] [0.83769339 0.21921516] [1639  508]
```

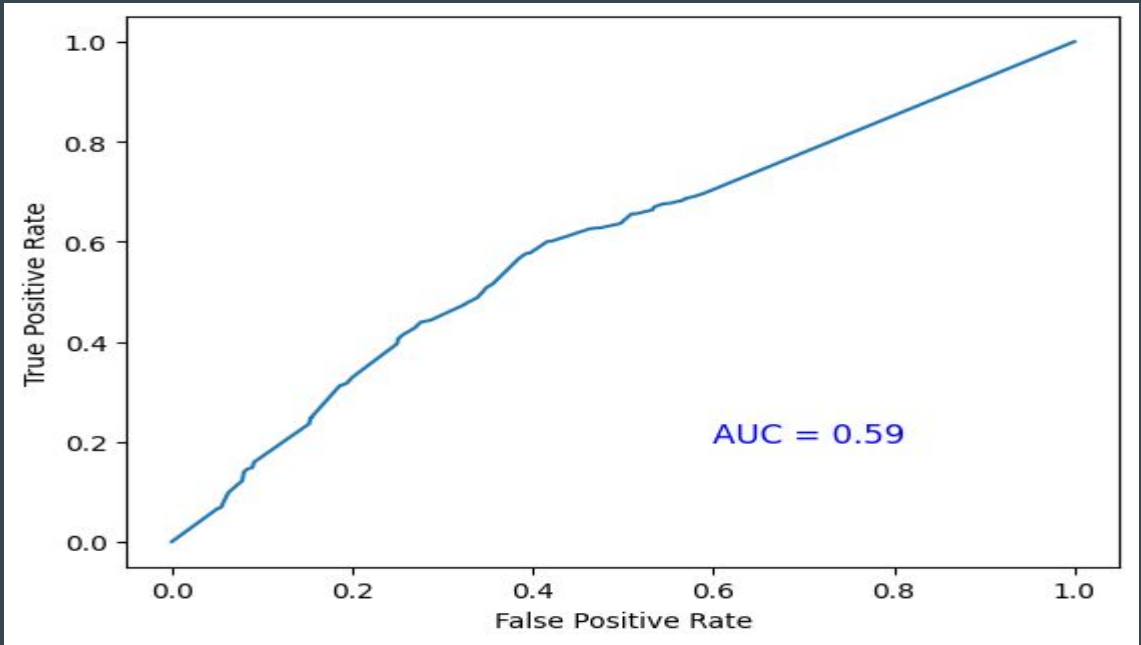
-Des faibles valeurs des métriques pour la classe “oui”, ceci est dû au déséquilibre entre les classes "non" et "oui".
-la faible précision pour la classe “oui” est due aux nombres élevés des False Positif.
-le faible rappel pour la classe “oui” est dû aux nombres élevés des False Négatif.

```
[ ] print(accuracy_score(Y_train, Y_p_train))
    print(accuracy_score(Y_test, Y_p_test))
```

```
0.8413806863527534
0.7312529110386586
```

```
[ ] df_train.to_csv('/content/drive/MyDrive/train_insurrance_preprocess.csv', index=False)
    df_test.to_csv('/content/drive/MyDrive/test_insurrance_preprocess.csv', index=False)
```

-Des bonnes valeurs de l'accuracy pour le data du train et du test, mais on peut pas compter seulement sur ces valeurs pour évaluer le modèle puisqu'il y a un déséquilibre entre les classes "non" et "oui".



-Le modèle semble avoir des difficultés à bien discriminer entre les classes, comme indiqué par l'AUC relativement basse.
-la valeur de l'AUC est supérieur à 0.5 , donc le modèle n'est pas médiocre.

Perspectives

Dans le but d'avoir un modèle plus performant et/ou des meilleures résultats on peut:
-essayer des modèles plus complexes.
-considérer des méthodes de gestion du déséquilibre de classe.
-en tenant compte des coûts associés aux faux négatifs et faux positifs, une analyse plus approfondie des conséquences financières et opérationnelles peut aider à orienter les ajustements du modèle.


MERCI DE VOTRE ATTENTION
makeameme.org