


```
import pandas as pd
df=pd.read_excel('titanic-passengers.xlsx')
df.head()
```



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
0	343	No	2	Collander, Mr. Erik Gustaf	male	28.0	0	0	248740	13.00
1	76	No	3	Moen, Mr. Sigurd Hansen	male	25.0	0	0	348123	7.65
2	641	No	3	Jensen, Mr. Hans Peder	male	20.0	0	0	350050	7.85
3	568	No	3	Palsson, Mrs. Nils (Alma	female	29.0	0	4	349909	21.07

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  -
3   Name        891 non-null    object
4   Sex         891 non-null    object
5   Age         714 non-null    float64
6   SibSp       891 non-null    int64
7   Parch       891 non-null    int64
8   Ticket      891 non-null    object
9   Fare        891 non-null    float64
10  Cabin       204 non-null    object
11  Embarked    889 non-null    object
dtypes: float64(2), int64(4), object(6)
memory usage: 83.7+ KB
```

```
print(df.isnull().sum().sum())
```

866

```
print(df.isnull().sum())
```

```
PassengerId    0
Survived        0
Pclass         0
Name           0
Sex            0
```

```
Age          177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin        687
Embarked      2
dtype: int64
```

```
df['Age'].fillna(df['Age'].mean(),inplace=True)
df.head(20)
```

Enregistrement...



29/05/2021

Untitled6.ipynb - Colaboratory

4	672	No	1	Davidson, Mr. Thornton	male	31.000000	1	0	F. 127
5	105	No	3	Gustafsson, Mr. Anders Vilhelm	male	37.000000	2	0	31012
6	576	No	3	Patchett, Mr. George	male	19.000000	0	0	3585
7	382	Yes	3	Nakid, Miss. Maria ("Mary")	female	1.000000	0	2	26
8	228	No	3	Lovell, Mr. John Hall ("Henry")	male	20.500000	0	0	F. 211
9	433	Yes	2	Louch, Mrs. Charles Alexander (Alice Adelaide ...)	female	42.000000	1	0	SC/ 30
10	135	No	2	Sobey, Mr. Samuel James Hayden	male	25.000000	0	0	C 291
11	294	No	3	Haas, Miss. Aloisia	female	24.000000	0	0	3492

Enregistrement... X

```

B96 B98      4
G6           4
F33          3
C22 C26      3
...
B80          1
F G63        1
D9           1
D47          1
C95          1
Name: Cabin, Length: 147, dtype: int64
```

Erland

31012

```
df['Cabin'].fillna('G6',inplace=True)
df['Cabin'].value_counts()
```

```

G6           691
C23 C25 C27    4
B96 B98        4
F33           3
C22 C26        3
...
B80           1
F G63         1
D9            1
D47           1
```

```
C95          1
Name: Cabin, Length: 147, dtype: int64
```

```
df['Embarked'].value_counts()
```

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

```
df['Embarked'].fillna('S',inplace=True)
df['Embarked'].value_counts()
```

```
S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

```
df.isnull().sum()
```

```
PassengerId    0
Survived        0
Pclass          0
Name            0
Sex             0
Age            0
SibSp           0
Parch           0
Ticket          0
```

Enregistrement...



```
dtype: int64
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
x = df[['PassengerId']]
y = df['Survived']
```

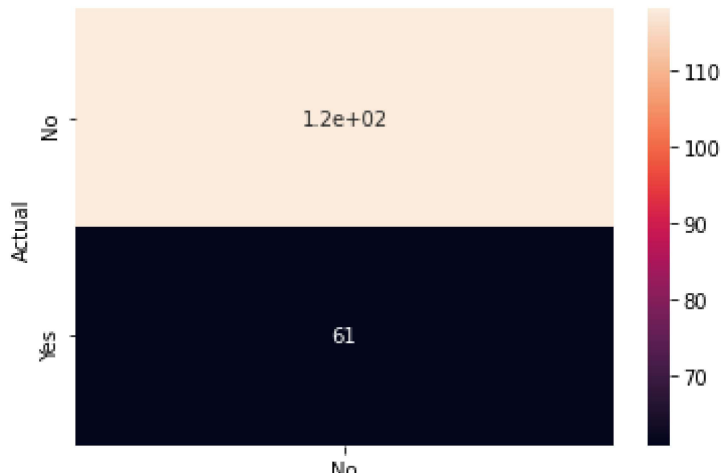
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

```
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
y_pred = logreg.predict(x_test)
print("Accuracy={:.2f}".format(logreg.score(x_test, y_test)))
```

```
Accuracy=0.66
```

```
import seaborn as sns
confusion_matrix = pd.crosstab(y_test, y_pred, rownames=['Actual'], colnames=['Predicted'])
sns.heatmap(confusion_matrix, annot=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f6ffbc2fc50>



Une courbe ROC (receiver operating characteristic) est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs en fonction du taux de faux positifs

AUC signifie "aire sous la courbe ROC". Cette valeur mesure l'intégralité de l'aire à deux dimensions située sous l'ensemble de la courbe ROC (par calculs d'intégrales) de (0,0) à (1,1).

```
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
```

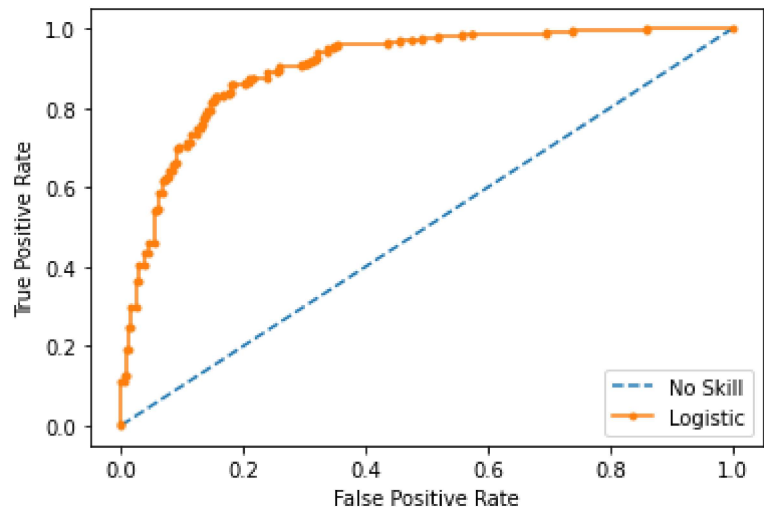
_score

Enregistrement...



```
X, y = make_classification(n_samples=1000, n_classes=2, random_state=1)
trainX, testX, trainy, testy = train_test_split(X, y, test_size=0.5, random_state=2)
ns_probs = [0 for _ in range(len(testy))]
model = LogisticRegression(solver='lbfgs')
model.fit(trainX, trainy)
lr_probs = model.predict_proba(testX)
lr_probs = lr_probs[:, 1]
ns_auc = roc_auc_score(testy, ns_probs)
lr_auc = roc_auc_score(testy, lr_probs)
print('No Skill: ROC AUC=%.3f' % (ns_auc))
print('Logistic: ROC AUC=%.3f' % (lr_auc))
# calculate roc curves
ns_fpr, ns_tpr, _ = roc_curve(testy, ns_probs)
lr_fpr, lr_tpr, _ = roc_curve(testy, lr_probs)
# plot the roc curve for the model
pyplot.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')
pyplot.plot(lr_fpr, lr_tpr, marker='.', label='Logistic')
# axis labels
pyplot.xlabel('False Positive Rate')
pyplot.ylabel('True Positive Rate')
# show the legend
pyplot.legend()
# show the plot
pyplot.show()
```

No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.903



Enregistrement... ✕