

Importing Library

```
import pandas as pd
import numpy as np
```

Choose dataset file from local directory

```
from google.colab import files
uploaded = files.upload()
```

Choose Files

train.csv

- **train.csv**(text/csv) - 61194 bytes, last modified: 3/30/2023 - 100% done

Saving train.csv to train (1).csv

Load Dataset

```
dataset = pd.read_csv('train.csv')
```

Summarize Dataset

```
print(dataset.shape)
print(dataset.head(5))
```

(891, 12)					
	PassengerId	Survived	Pclass	\	
0	1	0	3		
1	2	1	1		
2	3	1	3		
3	4	1	1		
4	5	0	3		
			Name	Sex	Age SibSp \
0			Braund, Mr. Owen Harris	male	22.0 1
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0		1
2	Heikkinen, Miss. Laina	female	26.0		0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0		1
4	Allen, Mr. William Henry	male	35.0		0
	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S

Mapping Text Data to Binary Value

```
income_set = set(dataset['Sex'])
dataset['Sex'] = dataset['Sex'].map({'female':0, 'male':1}).astype(int)
print(dataset.head())
```

<bound method NDFrame.head of				PassengerId	Survived	Pclass	\	
0	1	0	3					
1	2	1	1					
2	3	1	3					
3	4	1	1					
4	5	0	3					
..					
886	887	0	2					
887	888	1	1					
888	889	0	3					
889	890	1	1					
890	891	0	3					

				Name	Sex	Age	SibSp	\
0			Braund, Mr. Owen Harris	1	22.0	1		
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0		1			
2	Heikkinen, Miss. Laina	0	26.0		0			
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0		1			
4	Allen, Mr. William Henry	1	35.0		0			
..	
886	Montvila, Rev. Juozas	1	27.0		0			
887	Graham, Miss. Margaret Edith	0	19.0		0			
888	Johnston, Miss. Catherine Helen "Carrie"	0	NaN		1			

889			Behr, Mr. Karl Howell	1	26.0	0
890			Dooley, Mr. Patrick	1	32.0	0
	Parch	Ticket	Fare	Cabin	Embarked	
0	0	A/5 21171	7.2500	NaN	S	
1	0	PC 17599	71.2833	C85	C	
2	0	STON/O2. 3101282	7.9250	NaN	S	
3	0	113803	53.1000	C123	S	
4	0	373450	8.0500	NaN	S	
...	
886	0	211536	13.0000	NaN	S	
887	0	112053	30.0000	B42	S	
888	2	W./C. 6607	23.4500	NaN	S	
889	0	111369	30.0000	C148	C	
890	0	370376	7.7500	NaN	Q	

[891 rows x 12 columns]>

Segragate Dataset into X & Y

```
X = dataset.drop({'PassengerId', 'Name', 'Ticket', 'Cabin', 'Embarked', 'Survived'}, axis='columns')
X
```

	Pclass	Sex	Age	SibSp	Parch	Fare	
0	3	1	22.0	1	0	7.2500	
1	1	0	38.0	1	0	71.2833	
2	3	0	26.0	0	0	7.9250	
3	1	0	35.0	1	0	53.1000	
4	3	1	35.0	0	0	8.0500	
...	
886	2	1	27.0	0	0	13.0000	
887	1	0	19.0	0	0	30.0000	
888	3	0	NaN	1	2	23.4500	
889	1	1	26.0	0	0	30.0000	
890	3	1	32.0	0	0	7.7500	

891 rows x 6 columns

```
Y = dataset.Survived
Y
```

0	0
1	1
2	1
3	1
4	0
...	
886	0
887	1
888	0
889	1
890	0

Name: Survived, Length: 891, dtype: int64

Finding & Removing NA values from our Features X

```
X.columns[X.isna().any()]
```

Index(['Age'], dtype='object')

```
X.Age = X.Age.fillna(X.Age.mean())
```

Test again to check any NA value

```
X.columns[X.isna().any()]
```

Index([], dtype='object')

Splitting Dataset into Train & Test

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
```

Training

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)
```

▼ GaussianNB

GaussianNB()

Predicting for all Test Data

```
y_pred = model.predict(X_test)
print(np.column_stack((y_pred, y_test)))
```

```
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[1 1]
[0 0]
[0 0]
[1 1]
[0 0]
[1 1]
[0 0]
[0 0]
[1 0]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[1 1]
[0 0]
[0 0]
[1 1]
[1 0]
[0 0]
[1 1]
[1 1]
[0 0]
[1 0]
[0 0]
[1 1]
[1 1]
[0 0]
[0 1]
[1 0]
[0 0]
[1 1]
[0 1]
[0 0]
[1 0]
[0 0]
[1 1]
[0 0]
[0 0]
[0 1]
[0 0]
[1 0]
[0 0]
[0 0]
[0 1]
[1 0]
[1 1]
[0 0]
[1 1]
[1 1]
[1 1]]
```

Accuracy of our Model

```
from sklearn.metrics import accuracy_score  
print("Accuracy of the Model: {}".format(accuracy_score(y_test, y_pred)*100))
```

Accuracy of the Model: 78.02690582959642%

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 6:11 AM

