

```
In [7]: #Ayush Sharma 209303312
# 8.1 Program to demonstrate implementation of naïve Bayes Theorem in python
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
```

```
In [8]: url=r'https://raw.githubusercontent.com/codebasics/py/master/ML/14_naive_bayes/tita
titanicdf=pd.read_csv(url)
titanicdf.head()
```

```
Out[8]:
```

	PassengerId	Name	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
--	-------------	------	--------	-----	-----	-------	-------	--------	------	-------	----------

0	1	Braund, Mr. Owen Harris	3	male	22.0	1	0	A/5 21171	7.2500	NaN	
---	---	-------------------------------	---	------	------	---	---	--------------	--------	-----	--

1	2	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	female	38.0	1	0	PC 17599	71.2833	C85	
---	---	--	---	--------	------	---	---	----------	---------	-----	--

2	3	Heikkinen, Miss. Laina	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
---	---	------------------------------	---	--------	------	---	---	---------------------	--------	-----	--

3	4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	female	35.0	1	0	113803	53.1000	C123	
---	---	---	---	--------	------	---	---	--------	---------	------	--

4	5	Allen, Mr. William Henry	3	male	35.0	0	0	373450	8.0500	NaN	
---	---	--------------------------------	---	------	------	---	---	--------	--------	-----	--

```
In [9]: df=titanicdf
df.head()
```

Out[9]:

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

```
In [10]: df.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Cabin', 'Embarked'], axis=1)
df['Sex'] = pd.get_dummies(df['Sex'], drop_first=True)
df.head()
```

Out[10]:

	Pclass	Sex	Age	Fare	Survived
0	3	1	22.0	7.2500	0
1	1	0	38.0	71.2833	1
2	3	0	26.0	7.9250	1
3	1	0	35.0	53.1000	1
4	3	1	35.0	8.0500	0

```
In [11]: df.columns[df.isna().any()]
df.Age[:10]
df.Age = df.Age.fillna(df.Age.mean())
df.head()
```

Out[11]:

	Pclass	Sex	Age	Fare	Survived
0	3	1	22.0	7.2500	0
1	1	0	38.0	71.2833	1
2	3	0	26.0	7.9250	1
3	1	0	35.0	53.1000	1
4	3	1	35.0	8.0500	0

```
In [12]: X = df.drop(['Survived'], axis=1)
y = df['Survived']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_sta
```

```
In [13]: model = GaussianNB()
model.fit(X_train,y_train)
model.score(X_test,y_test)
```

Out[13]: 0.7761194029850746

```
In [14]: X_test[0:10]
y_test[0:10]
model.predict(X_test[0:10])
```

Out[14]: array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1], dtype=int64)

In []: