Naive Bayes Classifier (BEST Tutorial) :
==========

https://medium.com/datadriveninvestor/a-gentle-introduction-to-naive-bayes-classifier-9d7c4256c999

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)}$$

Bayes Theorem

- P(A|B) = **Posterior Probability,** Probability of A given Value of B.
- P (B|A) = **Likelihood** of B given A is True.
- P (A) = **Prior Probability,** Probability of event A.
- P (B) = **Marginal Probability,** Probability of event B.

By using the basis of the Bayes theorem, the Naive Bayes Classifier formula can be written as follows :

$$P\left(y \mid x_1, \cdots, x_j\right) = \frac{P\left(x_1, \cdots x_j \mid y\right) P(y)}{P\left(x_1, \cdots, x_j\right)}$$

Formula Naive Bayes Classifier

- P (y | x1, … , xj) = Posterior Probability, Probability of data included in class y given their features x1 until xj.
- P (x1, … , xj | y) = Likelihood of features value given that their class is y.
- P (y) = Prior Probability.
- P (x1, … , xj) = Marginal Probability.

2. Naive Bayes by Hands

Assume we have the following data in Table 1 and we want to classify new data with the following criteria :

- Age = **21–30**
- Income = **Medium**
- Status = **Married**

Age	Income	Status	Buy
<=20	low	students	yes
<=20	high	students	yes
<=20	medium	students	no
<=20	medium	married	no
<=20	high	married	yes
21 - 30	low	married	yes
21 - 30	low	married	no
21 - 30	medium	students	no
21 - 30	medium	married	no
21 - 30	high	students	yes
>30	high	married	no
>30	high	married	yes
>30	medium	married	yes
>30	medium	married	no
>30	low	students	no

Table 1. Data Training

1. First, Calculate **Prior Probability** P(y):
- *P (Buy=yes) → 7/15 → 0,467*
- P (Buy=no) → 8/15 → 0,533

2. Second, Calculate the **Likelihood** each features P (x | y):
- *P (Age = 21–30 | Buy=yes) → 2/7 → 0,285*
- P (Age = 21–30 | Buy=no) → 3/8 → 0,375
- *P (Income= Medium | Buy=yes) → 1/7 → 0,143*
- P (Income= Medium | Buy=no) → 5/8 → 0,625
- *P (Status= Married | Buy=yes) → 4/7 → 0,571*
- P (Status= Married | Buy=no) → 5/8 → 0,625

3. Calculate Total **Likelihood**
- *P (x | Buy=yes) → 0,285 * 0,143 * 0,571 = 0,0233*
- P (x | Buy=no) → 0,375 * 0,625 * 0,625 = 0,1464

4. Calculate **Posterior Probability** P(y | x):
- P (x | Buy=yes) * P (Buy=yes) → 0,0233 * 0,467 = 0,0108
- **P (x | Buy=no) * P (Buy=no) → 0,1464 * 0,533 = 0,0781**

From the above calculation, it was found that the new data belonged to the class that did **not buy.**

. . .

==========
==========

1 Load several libraries of python that will be used to work on this case.

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

import warnings
warnings.filterwarnings('ignore')
```

2 Load the dataset that will be used in working on this case. The dataset used is a glass dataset.

```
df = pd.read_csv('glass.csv')
```

6 Modeling our data with Gaussian Naive Bayes from Scikit-Learn.

```
# Create a Naive Bayes object
nb = GaussianNB()

#Create variable x and y.
x = df.drop(columns=['Type'])
y = df['Type']

#Split data into training and testing data
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=4)

#Training the model
nb.fit(x_train, y_train)

#Predict testing set
y_pred = nb.predict(x_test)

#Check performance of model
print(accuracy_score(y_test, y_pred))
```

```
0.4883720930232558
```

Accuracy Score

==========
==========