Intro. Artificial Intelligent
International University – VNU HCM
Dr. Ly Tu Nga

Course: IT097IU
Date: 27 March 2019
Time: 4 hours

Laboratory Session 2

# Naive Bayesian

## 1. Conditional Probability

Conditional probability is a measure of the probability of an event (some particular situation occurring) given that (by assumption, presumption, assertion or evidence) another event has occurred.

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$ → The conditional probability of A given that B has occurred

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}$$ → The conditional probability of B given that A has occurred

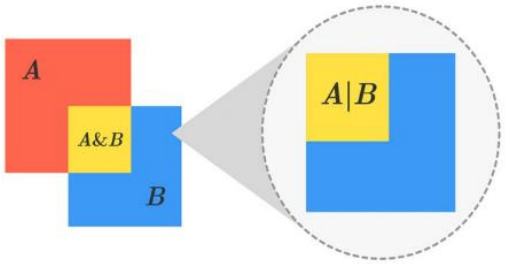
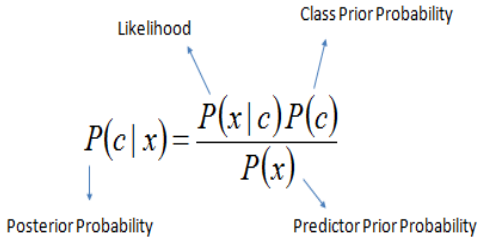



*Figure 1 Conditional probability*

## 2. Naive Bayesian Classifier [https://www.saedsayad.com/naive_bayesian.htm]

The Naive Bayesian classifier is based on Bayes' theorem with **the independence assumptions between predictors. A Naive Bayesian model is easy to build,** with no **complicated iterative parameter estimation which makes it particularly useful for very large datasets**. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.




$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

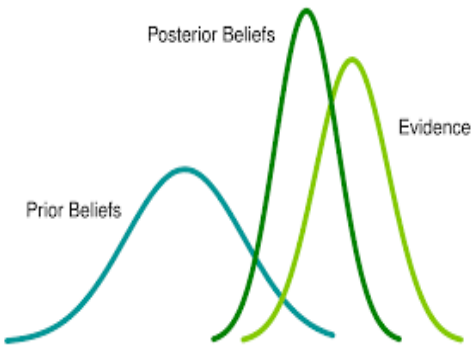



*Figure 2 Naïve Bayesian*

## 3. Random Variable Types

Qualitative Data: This type of data contains non-numerical values. The values taken by qualitative random variables are categorical in nature. Two broad classes or Qualitative Data:

- Nominal Data: The values represent categories which are not in any order.
  Example: Color of the flower: Red, blue, green
- Ordinal Data: The values represent categories which follow a particular order.
  Example: Income Class: Higher Income, Middle Income, Lower Income

Quantitative Data: The random variable takes numerical values. Two types of numerical values:

- Continuous: The variable can take any possible numerical value with in given bounds. Example: Height of a person: 5.5456 feet, 6 feet, 4.00001 feet
- Discrete: The variable can take only some values with given bounds. Example: Number of students in a class. The value can only be an integer.

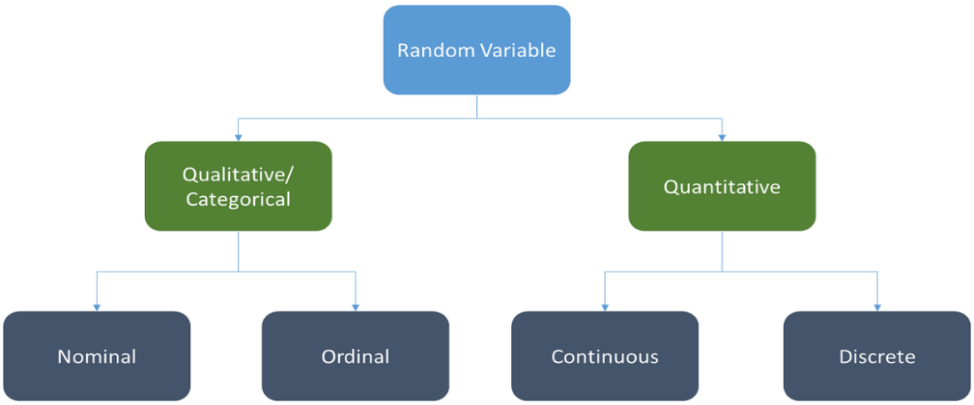



*Figure 3 Random variable types*

3. Practice

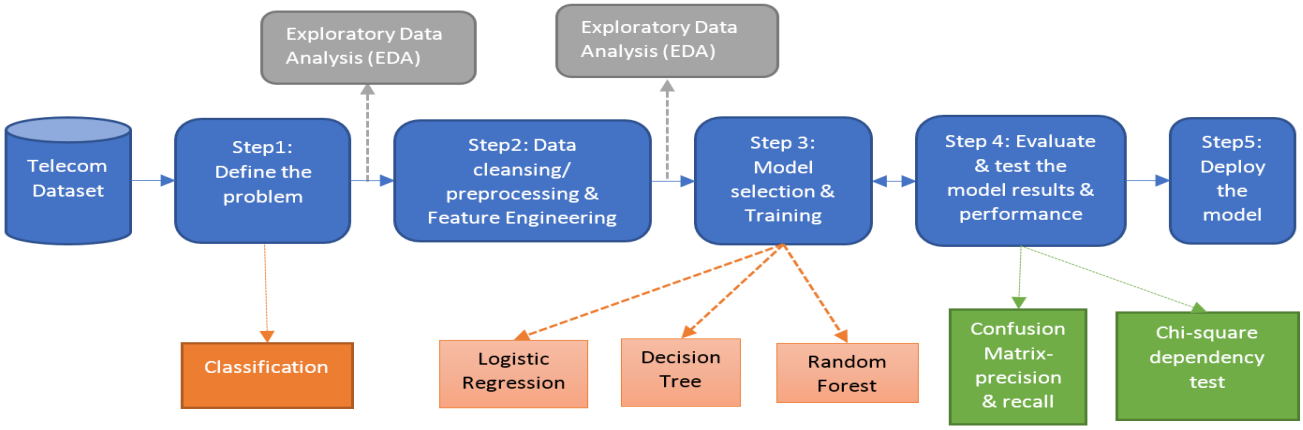



*Figure 4 The process of training models*

**3.1 Telecom Dataset:** 2 datasets: Customer Churn and Big Mart Chain

-Dataset1: Customer Churn is decried as follows

## Variable Description

| Name | Description | Value Type | Statistical Type |
|---|---|---|---|
| State | State Abbreviation (like KS – Kansas) | String | Categorical |
| Account length | How long the client has been with the company | Numerical | Quantitative |
| Area code | Phone number prefix | Numerical | Categorical |
| International plan | International Plan (on/off) | String. "Yes"/"No" | Categorical/Binary |
| Voice mail plan | Voice mail (on/off) | String. "Yes"/"No" | Categorical/Binary |
| Number vmail messages | Number of Voice mail messages | Numerical | Quantitative |
| Total day minutes | Total duration of daytime calls | Numerical | Quantitative |
| Total day calls | Total number of daytime calls | Numerical | Quantitative |
| Total day charge | Total charges for daytime services | Numerical | Quantitative |
| Total eve minutes | Total duration of evening calls | Numerical | Quantitative |
| Total eve calls | Total number of evening calls | Numerical | Quantitative |
| Total eve charge | Total charges for evening services | Numerical | Quantitative |
| Total night minutes | Total duration of nighttime calls | Numerical | Quantitative |
| Total night calls | Total number of nighttime calls | Numerical | Quantitative |
| Total night charge | Total charges for nighttime services | Numerical | Quantitative |
| Total intl minutes | Total duration of international calls | Numerical | Quantitative |
| Total intl calls | Total number of international calls | Numerical | Quantitative |
| Total intl charge | Total charges for international services | Numerical | Quantitative |
| Customer service calls | Number of calls to customer service | Numerical | Categorical/Ordinal |

-Dataset2: Big Mart Chain is to predict the price

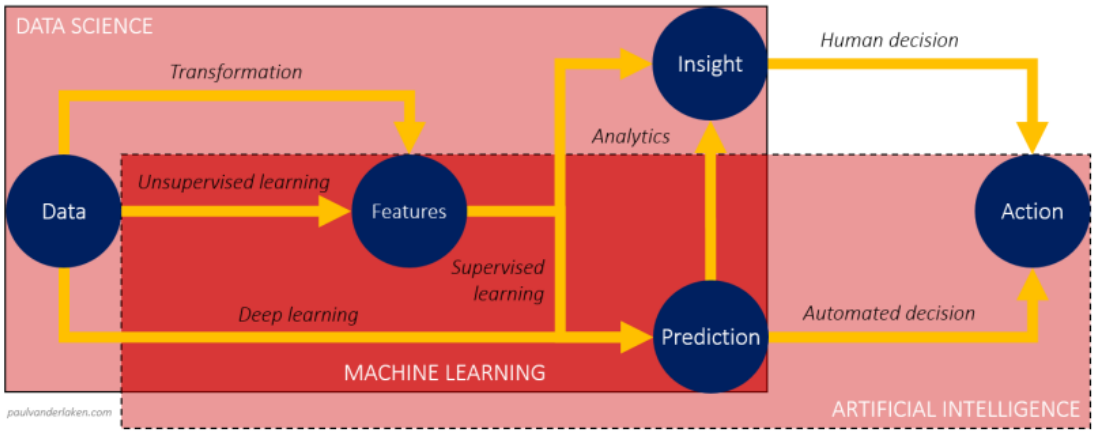



*Figure 5 AI process*

**3.2 Define the problem (Step1 of** Figure 4 The process of training models**)**

Given sales data for 1559 products across 10 stores of the Big Mart chain in various cities. The task is **to build a model to predict sales** for each particular product in different stores.

**The train and test data** contain the following variables shown in Figure 6.

| Variable | Description |
|---|---|
| Item_Identifier | Unique product ID |
| Item_Weight | Weight of product |
| Item_Fat_Content | Whether the product is low fat or not |
| Item_Visibility | % of total display area in store allocated to this product |
| Item_Type | Category to which product belongs |
| Item_MRP | Maximum Retail Price (list price) of product |
| Outlet_Identifier | Unique store ID |
| Outlet_Establishment_Year | Year in which store was established |
| Outlet_Size | Size of the store |
| Outlet_Location_Type | Type of city in which store is located |
| Outlet_Type | Grocery store or some sort of supermarket |
| Item_Outlet_Sales | Sales of product in particular store. This is the outcome variable to be predicted. |

*Figure 6 Variable and Description of dataset2*

-read 2 datasets (load the Read_data_csv.ipynb)
-Data cleansing [https://elitedatascience.com/data-cleaning]: check the percentage of missing values in each variable of all datasets. Hint use ***.isnull().sum()***
**Write down all variables that are missing data: _____**
**Choose the way to deal with missing data [read book, page 125], e.g. *fillna() based on nanmedian() [*https://www.geeksforgeeks.org/python-numpy-nanmedian-function/*] or mode() [*https://appdividend.com/2019/01/28/python-mode-function-example-python-statistics-tutorial/*]***

**3.3 Exploratory Data Analysis (EDA)**
**-use *.shape* to create the number of observations and attributes**

**3.4 Preprocessing**
Base on EDA and cleansing, p**lease collect at least 3 features (3 factors) that they have the high mean and std values. Hint use *.describe()***

| | | | |
|---|---|---|---|
| **Mean** | | | |
| **Std** | | | |

**-choose one of three features above to train. Hint, to avoid under fitting not choose the high mean and vice versa, to avoid over fitting not choose the high std.**
E.g. we choose the columns '*Total day minutes*', '*Churn*' to draw **histogram (hint use sns.distplot()** to plot the distribution with a kernel density estimate **[**http://seaborn.pydata.org/generated/seaborn.distplot.html?highlight=hist**]** and sns.jointplot()

After reset_index [https://pandas.pydata.org/pandas-docs/version/0.21/generated/pandas.DataFrame.reset_index.html] to draw a plot of two variables [http://seaborn.pydata.org/generated/seaborn.jointplot.html?highlight=reg]

**- collect at least 2 features (2 factors) that they have the high mean and std values and draw the histogram similar to the previous step.**

| | | |
|---|---|---|
| **Mean** | | |
| **Std** | | |

**-Use** *.State.unique()* **to category the 'State' of Customer Churn before find the relationship *'State'* and *'Churn'* (Hint** *groupby()* *[*https://www.kaggle.com/crawford/python-groupby-tutorial*]* **for 2 categories above)**

```
List of unique values in State :
['KS' 'OH' 'NJ' 'OK' 'AL' 'MA' 'MO' 'LA' 'WV' 'IN' 'RI' 'IA' 'MT' 'NY'
 'ID' 'VT' 'VA' 'TX' 'FL' 'CO' 'AZ' 'SC' 'NE' 'WY' 'HI' 'IL' 'NH' 'GA'
 'AK' 'MD' 'AR' 'WI' 'OR' 'MI' 'DE' 'UT' 'CA' 'MN' 'SD' 'NC' 'WA' 'NM'
 'NV' 'DC' 'KY' 'ME' 'MS' 'TN' 'PA' 'CT' 'ND']
```

| | State | Churn | Freq |
|---|---|---|---|
| 1 | AL | False | 72 |
| 2 | VA | False | 72 |
| 3 | WI | False | 71 |
| 4 | MN | False | 69 |
| 5 | NY | False | 68 |
| 6 | WY | False | 68 |
| 7 | OH | False | 68 |
| 8 | OR | False | 67 |
| 9 | VT | False | 65 |
| 10 | ID | False | 64 |

**-find the relationship 'Account length >100' and 'Churn'=False. And create a table with columns State, Account length and Churn.**

| | State | Account length | Churn |
|---|---|---|---|
| 1 | OH | 107 | False |
| 2 | NJ | 137 | False |
| 5 | AL | 118 | False |
| 6 | MA | 121 | False |
| 7 | MO | 147 | False |
| 8 | LA | 117 | False |
| 9 | WV | 141 | False |

Data Visualization (EDA)

-Draw the Barplot of 'State'

-Draw the Barplot of 'Churn'

-Draw the Countplot for 'Area code' and 'Churn'

-Draw the Countplot for 'Customer service calls' and 'Churn'

-Draw the histogram for 'Account length'

-Draw the Boxplot for 'Total day minutes'

-Print out the statistic of ‘Total day minutes’

```
 Statistics: Summary of Total day minutes
count    3333.000000
mean      179.775098
std        54.467389
min         0.000000
25%       143.700000
50%       179.400000
75%       216.400000
max       350.800000
Name: Total day minutes, dtype: float64
```

-Draw Boxplot for ‘Churn’ and ‘Total day minutes’
-Draw Boxplot for ‘Churn’, ‘Total day minutes’ and ‘International plan’
**-Draw the PieChart with’ Area Code’**

## Data Encoding

Before data encoding, create a new classification dataframe *ClassificationDF* of Orginal CustomerChurn (Hint use copy()).
-**encode categorical features** using a **one-hot scheme** (Hint use .fit _transform()) [https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html] for ‘State’, ‘International plan’, ‘Voice mail plan’ and ‘Area_code’
Prediction: create a new prediction dataframe with dropped ‘Item_Identifier’ and ‘Outlet_Identifier’

| Outlet_Location_Type_Tier 3 | Outlet_Type_Grocery Store | Outlet_Type_Supermarket Type1 | Outlet_Type_Supermarket Type2 | Outlet_Type_Supermarket Type3 |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |

## Training preparation

-create a random training data with the rate is the entire set [https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sample.html] Hint use *DataFrame.sample()*
-create a testing set with the left out portion of the dataset. Hint use *DataFram.loc [https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.loc.html]*
-print out the training rows and columns, test rows and columns values

**-Classification Process via Naïve Bayes Learning/Testing**

```
def NaiveBayesLearning(DataTrain, TargetTrain):
    from sklearn.naive_bayes import GaussianNB
    NBModel = GaussianNB()
    NBModel.fit(DataTrain, TargetTrain.values.ravel())

    return NBModel

def NaiveBayesTesting(NBModel,DataTest, TargetTest):
    from sklearn.metrics import accuracy_score
    PredictTest = NBModel.predict(DataTest)
    Accuracy = accuracy_score(TargetTest, PredictTest)

    return Accuracy, PredictTest
```

-Show the predict probability of Testing

```
[ ] NBModel = NaiveBayesLearning(CX_train, Cy_train)
    NBAccuracy,PredictTest = NaiveBayesTesting(NBModel,CX_test, Cy_test)
    print('Naive Bayes accuracy: {:.3f}'.format(NBAccuracy))

    Naive Bayes accuracy: 0.867

[ ] NBModel.get_params

    <bound method BaseEstimator.get_params of GaussianNB(priors=None, var_smoothing=1e-09)>

[ ] NBModel.predict_proba(CX_train)[1,:]

    array([0.95467315, 0.04532685])
```

4. Repeat 3.4 for other 2 features

5. Make conclusions about 3 features