UIC
BUSINESS

Business Data Mining

IDS 472 (Spring 2024)

Instructor: Wenxin Zhou

Naïve Bayes Classifier

UIC BUSINESS

- This chapter introduces the naïve Bayes (NB) classifier, applicable to data with categorical predictors
- Start with the concepts of
  - conditional probability
  - complete, or exact, Bayesian classifier
  - pros and cons
- Modify the Bayesian classifier to define the NB classifier
- We will use `pandas` for data handling, `scikit-learn` for naïve Bayes models, and `matplotlib` for visualization

import required functionality for this chapter

```
import pandas as pd
import matplotlib.pylab as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix
```

Bayesian Classifier

UIC BUSINESS

To understand NB classifier, we first look at the complete/exact Bayesian classifier

find all the other records with the same predictor profile (predictor values are the same)	determine what classes the records belong to and which class is most prevalent	assign that class to the new record

Tweak the method to answer the questions:

"What is the propensity of belonging to the class of interest?"

instead of

"Which class is the most probable?"

Cutoff Probability Method

UIC BUSINESS

Establish a cutoff probability for the class of interest above which we consider that a record belongs to that class

1

Find all the training records with the same predictor profile as the new record

2

Determine the probability that those records belong to the class of interest

3

If that probability is above the cutoff probability, assign the new record to the class of interest

Conditional Probability

UIC BUSINESS

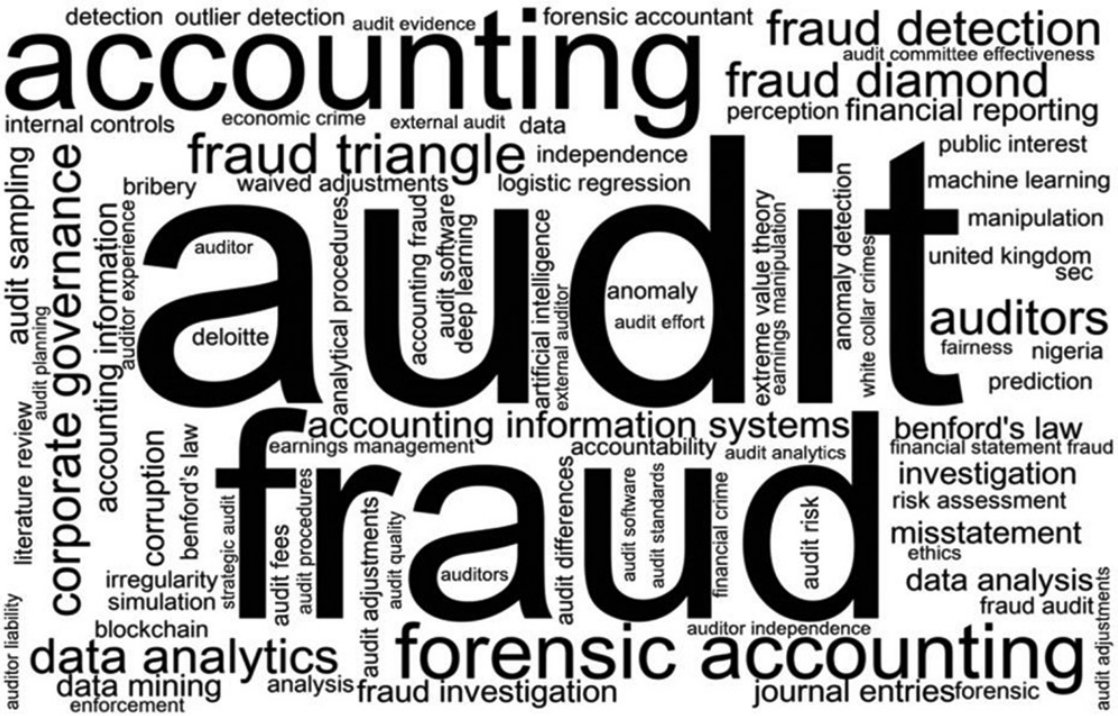
- Conditional probability: the probability of event A given that event B has occurred, denoted $P(A|B)$
- We will be looking at the probability of a record belonging to class C_i given that its predictor values are $x_1, \ldots, x_p$
- For a response with m classes $C_1, \ldots, C_m$, and predictor values $x_1, \ldots, x_p$, we want to compute

$$P(C_i|x_1, \ldots, x_p)$$

- Classify the record
  1) to the class that has the highest probability
  2) using the cutoff probability to decide whether it should be assigned to the class of interest
- Bayesian classifier works only with categorical predictors: it is the ONLY classification method presented in this class that is especially suited for categorical predictors

Example 1

UIC BUSINESS



- Question: whether a customer has a history of legal troubles, including criminal or civil charges, can be predictive of fraudulent reporting

Example 1

UIC BUSINESS

- Each customer is a record, outcome $Y = \{C_1 = \text{fraudulent}, C_2 = \text{truthful}\}$, predictor takes two values: 0 (no prior legal trouble) and 1 (prior legal trouble)
- The accounting firm has data on 1500 companies
- For each company, it has information on whether the financial report was judged fraudulent or truthful and whether the company had prior legal trouble
- Partition the data into a training set (1000 firms) and a validation set (500 firms)

	Prior legal ($X = 1$)	No prior legal ($X = 0$)	Total
Fraudulent (C_1)	50	50	100
Truthful (C_2)	180	720	900
Total	230	770	1000

Full Bayesian Classifier

UIC BUSINESS

- If a new company had had prior legal trouble, the probability of belonging to the "F" class is $P(\text{fraudulent} \mid \text{prior legal}) = \frac{50}{230}$; therefore, $P(\text{truthful} \mid \text{prior legal}) = \frac{180}{230}$
- "Assign to the Most Probable Class" Method:
  - if a company had prior legal trouble, assign it to the "T" class
  - if no prior legal trouble, also assign it to the "T" class
  - all records are assigned to the "T" class
  - naive rule of "assign all records to the majority class"
- Cutoff Probability Method:
  - to identify fraudulent reports, some truthful reports will be misidentified, and the overall classification accuracy may decline
  - establish a cutoff value (for the probability of being F), and classify all records above that value as F

Full Bayesian Classifier

UIC BUSINESS

- Bayesian formula for the probability that a record belongs to class C_i is

$$P(C_i|x_1,\ldots,x_p) = \frac{P(x_1,\ldots,x_p|C_i)P(C_i)}{P(x_1,\ldots,x_p|C_1)P(C_1)+\cdots+P(x_1,\ldots,x_p|C_m)P(C_m)}$$

- In this example (frauds are rarer), if the cutoff were set to 0.2, we would classify a prior legal trouble record as fraudulent because $\frac{50}{230} = 0.22$

- The user can treat this cutoff as a "slider" to be adjusted to optimize performance

Practical Issue

UIC BUSINESS

- To estimate the probability $P(C|x_{\text{new}})$, we need to find all the records i s.t. $x_i = x_{\text{new}}$
- When the number of predictors – $\dim(x_i)$ – gets larger, many of the records to be classified will be without exact matches
- For example, even a sizable sample may not contain even a single match for a new record who is
  - male
  - Hispanic
  - from US Midwest
  - high income
  - voted in the last election
  - did not vote in the prior election
  - three daughters and one son
  - divorced

Naïve Bayes

UIC BUSINESS

- From the Bayesian formula, the main difficulty in estimating $P(C_i|x_1, \dots, x_p)$ comes from the estimation of $P(x_1, \dots, x_p|C_i)$

- The Naïve Bayes Assumption of Conditional Independence: predictors are independent within each class

$$P(x_1, \dots, x_p|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_p|C_i)$$

- If $x_1 =$ "prior legal trouble" and $x_2 =$ "lost money last year", within a given class, we no longer need to look for records characterized by both; rather, we simply multiply the probability of "prior legal trouble" by the probability of "lost money last year"

Naïve Bayes

UIC BUSINESS

1. For class C_i, define $\hat{P}(x_j|C_i)$ as the frequency of C_i-labelled records (in the training set) whose j^{th} feature equals x_j

2. Let $\hat{P}(C_i)$ be the proportion of records belonging to class C_i

3. Estimate $P(C_i|x_1, \ldots, x_p), i = 1, \ldots, m$ by

$$\hat{P}_{nb}(C_i|x_1, \ldots, x_p) = \frac{\hat{P}(C_i)\hat{P}(x_1|C_i) \cdots \hat{P}(x_p|C_i)}{\hat{P}(C_1)\hat{P}(x_1|C_1) \cdots \hat{P}(x_p|C_1) + \cdots + \hat{P}(C_m)\hat{P}(x_1|C_m) \cdots \hat{P}(x_p|C_m)}$$

4. Assign the record to the class with the highest probability for this set of predictor values, or use the cutoff probability method

Example 2

UIC BUSINESS

- Expand the financial reports example to two predictors
- For each customer, we have information on whether it had prior legal trouble, whether it is a small/large company and whether the financial report was found to be fraudulent/truthful

Company	Prior legal trouble	Company size	Status
1	Yes	Small	Truthful
2	No	Small	Truthful
3	No	Large	Truthful
4	No	Large	Truthful
5	No	Small	Truthful
6	No	Small	Truthful
7	Yes	Small	Fraudulent
8	Yes	Large	Fraudulent
9	No	Large	Fraudulent
10	Yes	Large	Fraudulent

Example 2

UIC BUSINESS

- Complete (Exact) Bayes Calculations:

$$P(\text{fraudulent}|\text{PriorLegal} = \text{y}, \text{Size} = \text{small}) = 1/2 = 0.5$$
$$P(\text{fraudulent}|\text{PriorLegal} = \text{y}, \text{Size} = \text{large}) = 2/2 = 1$$
$$P(\text{fraudulent}|\text{PriorLegal} = \text{n}, \text{Size} = \text{small}) = 0/3 = 0$$
$$P(\text{fraudulent}|\text{PriorLegal} = \text{n}, \text{Size} = \text{large}) = 1/3 = 0.33$$

- Naïve Bayes Calculations:

$$P_{\text{nb}}(\text{fraudulent}|\text{PriorLegal} = \text{y}, \text{Size} = \text{small}) = \frac{(3/4)(1/4)(4/10)}{(3/4)(1/4)(4/10) + (1/6)(4/6)(6/10)} = 0.53$$
$$P_{\text{nb}}(\text{fraudulent}|\text{PriorLegal} = \text{y}, \text{Size} = \text{large}) = 0.87$$
$$P_{\text{nb}}(\text{fraudulent}|\text{PriorLegal} = \text{n}, \text{Size} = \text{small}) = 0.07$$
$$P_{\text{nb}}(\text{fraudulent}|\text{PriorLegal} = \text{n}, \text{Size} = \text{large}) = 0.31$$

Predicting Delayed Flights

UIC BUSINESS

- Predicting flight delays can be useful to airport authorities, airlines, and aviation authorities
- In this example, we have five predictors

Day of week	Coded as 1 = Monday, 2 = Tuesday, ..., 7 = Sunday
Sch. dep. time	Broken down into 18 intervals between 6:00 AM and 10:00 PM
Origin	Three airport codes: DCA (Reagan National), IAD (Dulles), BWI (Baltimore–Washington Int'l)
Destination	Three airport codes: JFK (Kennedy), LGA (LaGuardia), EWR (Newark)
Carrier	Eight airline codes: CO (Continental), DH (Atlantic Coast), DL (Delta), MQ (American Eagle), OH (Comair), RU (Continental Express), UA (United), and US (USAirways)

- Outcome: whether the flight is delayed (>15min late)
- The data consist of all flights from DC into NYC in 01/2004
- There are 2201 records, each denotes a particular flight
- The percentage of delayed flights is 19.5%

Predicting Delayed Flights

UIC BUSINESS

- Goal: predict whether a new flight will be delayed (1 = delayed and 0 = on time)
- Implementation process:
  1. convert all predictors to categorical and creating dummies
  2. partition data into training (60%) and validation (40%) sets
  3. apply a naive Bayes classifier to the training set

- To see how the algorithm works, we generate pivot tables for the outcome vs. each of the five predictors using the training set, in order to obtain conditional probabilities

- To classify a new flight, compute its delay probability and on-time probability (only the numerators)

Predicting Delayed Flights

UIC BUSINESS

code for running naive Bayes

```
delays_df = pd.read_csv('FlightDelays.csv')

# convert to categorical
delays_df.DAY_WEEK = delays_df.DAY_WEEK.astype('category')
delays_df['Flight Status'] = delays_df['Flight Status'].astype('category')

# create hourly bins departure time
delays_df.CRS_DEP_TIME = [round(t / 100) for t in delays_df.CRS_DEP_TIME]
delays_df.CRS_DEP_TIME = delays_df.CRS_DEP_TIME.astype('category')

predictors = ['DAY_WEEK', 'CRS_DEP_TIME', 'ORIGIN', 'DEST', 'CARRIER']
outcome = 'Flight Status'

X = pd.get_dummies(delays_df[predictors])
y = delays_df['Flight Status'].astype('category')
classes = list(y.cat.categories)

# split into training and validation
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.40,
                                                      random_state=1)

# run naive Bayes
delays_nb = MultinomialNB(alpha=0.01)
delays_nb.fit(X_train, y_train)

# predict probabilities
predProb_train = delays_nb.predict_proba(X_train)
predProb_valid = delays_nb.predict_proba(X_valid)

# predict class membership
y_valid_pred = delays_nb.predict(X_valid)
```

Predicting Delayed Flights

UIC BUSINESS

- Pivot table of flight status by destination airport (training data)

```
# split the original data frame into a train and test using the same random_state
train_df, valid_df = train_test_split(delays_df, test_size=0.4, random_state=1)

pd.set_option('precision', 4)
# probability of flight status
print(train_df['Flight Status'].value_counts() / len(train_df))
print()

for predictor in predictors:
    # construct the frequency table
    df = train_df[['Flight Status', predictor]]
    freqTable = df.pivot_table(index='Flight Status', columns=predictor, aggfunc=len)

    # divide each value by the sum of the row to get conditional probabilities
    propTable = freqTable.apply(lambda x: x / sum(x), axis=1)
    print(propTable)
    print()
pd.reset_option('precision')
```

Output

Pivot Table

UIC BUSINESS

```
ontime     0.8023
delayed    0.1977

DAY_WEEK            1       2       3       4       5      6       7
Flight Status
delayed        0.1916  0.1494  0.1149  0.1264  0.1877  0.069  0.1609
ontime         0.1246  0.1416  0.1445  0.1794  0.1690  0.136  0.1048

CRS_DEP_TIME        6       7       8       9       10      11      12      13  \
Flight Status
delayed        0.0345  0.0536  0.0651  0.0192  0.0307  0.0115  0.0498  0.0460
ontime         0.0623  0.0633  0.0850  0.0567  0.0519  0.0340  0.0661  0.0746

CRS_DEP_TIME       14      15      16      17      18      19      20      21
Flight Status
delayed        0.0383  0.2031  0.0728  0.1533  0.0192  0.0996  0.0153  0.0881
ontime         0.0576  0.1171  0.0774  0.1001  0.0349  0.0397  0.0264  0.0529

ORIGIN             BWI     DCA     IAD
Flight Status
delayed        0.0805  0.5211  0.3985
ontime         0.0604  0.6478  0.2918

DEST               EWR     JFK     LGA
Flight Status
delayed        0.3793  0.1992  0.4215
ontime         0.2663  0.1558  0.5779

CARRIER             CO      DH      DL      MQ      OH      RU      UA      US
Flight Status
delayed        0.0575  0.3142  0.0958  0.2222  0.0077  0.2184  0.0153  0.0690
ontime         0.0349  0.2295  0.2040  0.1171  0.0104  0.1690  0.0170  0.2181
```

Frequency Table

UIC BUSINESS

```
day_week          1    2    3    4    5    6    7
flight_status
delayed          50   39   30   33   49   18   42
ontime          132  150  153  190  179  144  111

crs_dep_time     6   7   8   9  10  11  12  13  14   15  16   17  18  19  20  \
flight_status
delayed          9  14  17   5   8   3  13  12  10   53  19   40   5  26   4
ontime          66  67  90  60  55  36  70  79  61  124  82  106  37  42  28

crs_dep_time    21
flight_status
delayed         23
ontime          56

origin          BWI  DCA  IAD
flight_status
delayed          21  136  104
ontime           64  686  309

dest            EWR  JFK  LGA
flight_status
delayed          99   52  110
ontime          282  165  612

carrier         CO   DH   DL   MQ  OH   RU  UA   US
flight_status
delayed         15   82   25   58   2   57   4   18
ontime          37  243  216  124  11  179  18  231
```

Prediction

UIC BUSINESS

- To classify a Delta flight from DCA to LGA departing between 10:00 AM and 11:00 AM on a Sunday, we compute

$$\hat{P}(\text{delayed}|\text{Carrier} = \text{DL}, \text{Day\_Week} = 7, \text{Dep\_Time} = 10, \text{Dest} = \text{LGA}, \text{Origin} = \text{DCA})$$
$$\propto (0.0958)(0.1609)(0.0307)(0.4215)(0.5211)(0.2) = 0.000021,$$
$$\hat{P}(\text{ontime}|\text{Carrier} = \text{DL}, \text{Day\_Week} = 7, \text{Dep\_Time} = 10, \text{Dest} = \text{LGA}, \text{Origin} = \text{DCA})$$
$$\propto (0.2040)(0.1048)(0.0519)(0.5779)(0.6478)(0.8) = 0.00033.$$

  The symbol $\propto$ means "is proportional to".

- A record with such a combination of predictor values does not exist in the training set (but does exist in the test set)
- To compute the actual probability,

$$\hat{P}(\text{delayed}|\text{Carrier} = \text{DL}, \text{Day\_Week} = 7, \text{Dep\_Time} = 10, \text{Dest} = \text{LGA}, \text{Origin} = \text{DCA})$$
$$= \frac{0.000021}{0.000021 + 0.00033} = 0.058,$$
$$\hat{P}(\text{on time}|\text{Carrier} = \text{DL}, \text{Day\_Week} = 7, \text{Dep\_Time} = 10, \text{Dest} = \text{LGA}, \text{Origin} = \text{DCA})$$
$$= \frac{0.00033}{0.000021 + 0.00033} = 0.942.$$

Confusion Matrix

UIC BUSINESS

```
from sklearn.metrics import confusion_matrix

cm_train = pd.DataFrame(confusion_matrix(y_train, y_train_pred), index=classes, columns=classes)
print('Confusion Matrix (train):')
print(cm_train)
print('Accuracy: {:.4f}'.format(accuracy_score(y_train, y_train_pred)), end='\n\n')

cm_valid = pd.DataFrame(confusion_matrix(y_valid, y_valid_pred), index=classes, columns=classes)
print('Confusion Matrix (valid):')
print(cm_valid)
print('Accuracy: {:.4f}'.format(accuracy_score(y_valid, y_valid_pred)))
```

✓ 0.0s Python

```
Confusion Matrix (train):
         delayed  ontime
delayed       52     209
ontime        61     998
Accuracy: 0.7955

Confusion Matrix (valid):
         delayed  ontime
delayed       26     141
ontime        51     663
Accuracy: 0.7821
```

Summary

UIC BUSINESS

- The NB classifier's beauty is in its simplicity, computational efficiency, good classification performance, and ability to handle categorical variables directly
- Three main issues:
  1) Need a very large number of records to obtain good results
  2) When a predictor category is not present in the training data, NB assumes that a new record with that category of the predictor has zero probability
  3) Good performance is obtained when the goal is *classification* or *ranking* of records according to their probability of belonging to a certain class. When the goal is to *estimate the probability of class membership* (propensity), this method provides very biased results. For this reason, the NB method is rarely used in credit scoring.