Analytical information systems
Naive Bayes Algorithm

Name: Tanayraj Jhagadiawala

Matrikel Nr.: 00779019

Instructor's Name: Prof. Dr. Gefei Zhang

hochschule
hof

Introduction: What is Naive Bayes Algorithm?

You are working on a classification problem and have generated your set of hypothesis, created features and discussed the importance of variables. Within an hour, stakeholders want to see the first cut of the model.

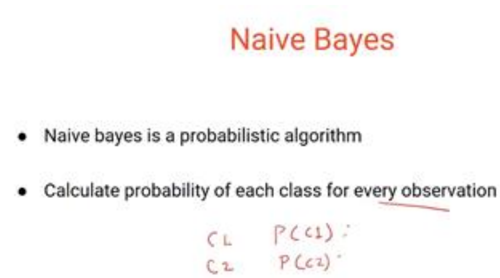
What will you do? You have hundreds of thousands of data points and quite a few variables in your training data set. In such a situation, if I were in your place, I would have used ‘Naive Bayes‘, which can be extremely fast relative to other classification algorithms. It works on Bayes theorem of probability to predict the class of unknown data sets.

In statistics, Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models. But they could be coupled with Kernel density estimation and achieve higher accuracy levels.

Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting.



If we have 2 classes, c1 and c2. For every class Naïve Bayes calculate

Probability of that observation comes in class 1

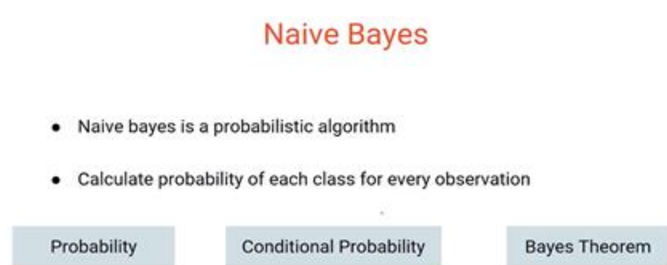
Probability of that observation comes in class 2

Whichever has a higher probability will be selected as a class to observation or main focus.

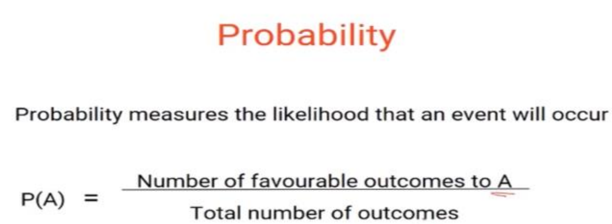
Part of Naïve Bayes

We can divide Naïve Bayes algorithm into 3 parts

1. Probability
2. Conditional Probability
3. Bayes Theorem



1. Probability



If probability of event is near to zero then we can say chances of occurs that event is very low. On the other hand if probability of event is near to 1, for example 0.9 than we can say chances of occurs that event is very high.

Let’s try to understand with on example. There is one dice and when we roll the dice we want to check probability of even number appear on dice. Here total number of output is 6. 1, 2, 3, 4, 5, 6. And total even number are three. 2, 4, 6. So now I put value in the equation so 3/6 is 0.5. So probability of even number appear on dice is 0.5.

2. Conditional probability

The conditional probability of an event B is the probability that the event will occur given the knowledge that an event A has already occurred. This probability is written P(B|A), notation for the probability of B given A. In the case where events A and B are independent (where event A has no effect on the probability of event B), the conditional probability of event B given event A is simply the probability of event B, that is P(B).

If events A and B are not independent, then the probability of the intersection of A and B (the probability that both events occur) is defined by P(A and B) = P(A)P(B|A).

From this definition, the conditional probability P(B|A) is easily obtained by dividing by P(A):

$$P(B|A) = \frac{P(A\ and\ B)}{P(A)}$$

Conditional Probability

Calculate the probability of the an event for a given condition

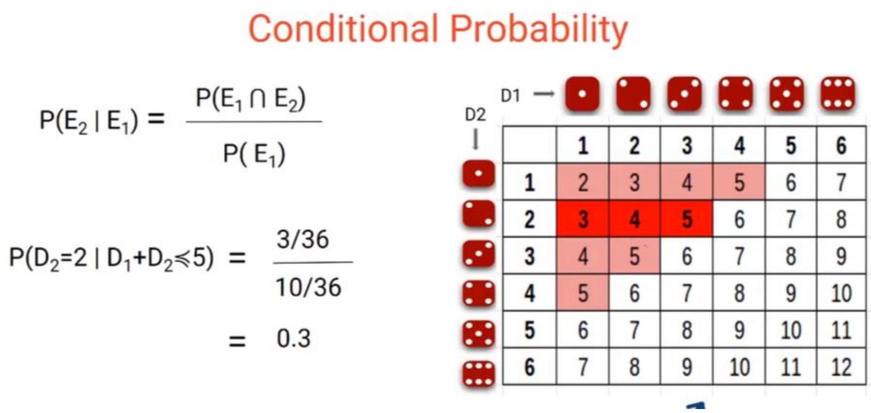
$P(\,\underbrace{D_2=2}\,|\,\underbrace{D_1 + D_2 \leqslant 5}\,)$

Second dice gave value of 2 and sum of two dices should be less than or equal to 5. To calculate Probability of occurrence of event for given condition, I divide above equation into the two events.

$P(\,D_2=2 \mid D_1 + D_2 \leqslant 5)$

Event 1: $D_1 + D_2 \leqslant 5$

Event 2: $D_2=2$



Now let's try to understand with 6*6 table. The Probability of Event 1 will occur is highlighted in pink and The Probability of Event 2 will occur is highlighted in red. So we can see there are only three possibility which can fulfill both the event. It can be said as intersection of two event. So it's like out 36 possibility only 3 possibility can fulfill both the conditions. Now we put value into the equation and we get 0.3. So probability of Second dice should value of 2 and sum of two dices should be less than or equal to 5 is 0.3.

3. **Bayes Theorem**

Bayes' theorem is stated mathematically as the following equation:

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

Where A and B are events and P(B)!= 0.

P(A\ B) is a conditional probability: the likelihood of event A occurring given that B is true.

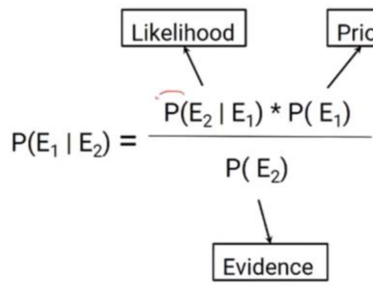
P(B\ A) is also a conditional probability: the likelihood of event B occurring given that A is true.

P(A) and P(B) are the probabilities of observing A and B respectively; they are known as the marginal probability.

A and B must be different events.

Using Bayes' theorem, the conditional probability can be decomposed as

Bayes Theorem



Concept behind Naïve Bayes

$$P(C_k \mid X) = \frac{P(X \mid C_k) * P(C_k)}{P(X)}$$

P(Ck|X) is more generalized equation. The likelihood of event Ck occurring given that X is true.
Now use this equation to solve below example.

Concept behind Naive Bayes

Outlook	Temp	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

$$X = [\underbrace{\text{Outlook}}_{x_1}, \underbrace{\text{Temp}}_{x_2}, \underbrace{\text{Humidity}}_{x_3}, \underbrace{\text{Windy}}_{x_4}]$$

$$C_k = [\underbrace{\text{Yes}}_{C_1}, \underbrace{\text{No}}_{C_2}]$$

Here, we want to predict that person or team will play or not in these conditions.

X = features like Outlook, Temperature, Humidity, Windy

Ck= class like Yes or No

Concept behind Naive Bayes

Outlook	Temp	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

$$P(E_1 \mid E_2) = \frac{P(E_2 \mid E_1) * P(E_1)}{P(E_2)}$$

$$P(C_k \mid X) = \frac{P(X \mid C_k) * P(C_k)}{P(X)}$$

Concept behind Naive Bayes

$$P(C_k \mid X) = \frac{P(X \mid C_k) * P(C_k)}{P(X)}$$

$$P(C_1 \mid x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \cap x_2 \cap x_3 \cap x_4 \mid C_1) * P(C_1)}{P(x_1 \cap x_2 \cap x_3 \cap x_4)}$$

$$\boxed{P(C_1 \mid x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \mid C_1) * P(x_2 \mid C_1) * P(x_3 \mid C_1) * P(x_4 \mid C_1) * P(C_1)}{P(x_1) * P(x_2) * P(x_3) * P(x_4)}}$$

Now we simplify this equation. Question might occur why we take intersection. It’s simple, we want to check probability when above every condition is happing at same time. Naïve Bayes assume all the features are independent from each other. We father simplify equation.

Now we check for both the class, person will play or not based on given features. Whichever has higher probability will be taken as final class.

Concept behind Naive Bayes

$$\boxed{P(C_1 \mid x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \mid C_1) * P(x_2 \mid C_1) * P(x_3 \mid C_1) * P(x_4 \mid C_1) * P(C_1)}{P(x_1) * P(x_2) * P(x_3) * P(x_4)}}$$

$$P(yes \mid sunny \cap hot \cap high \cap windy) = \frac{P(sunny \mid yes) * P(hot \mid yes) * P(high \mid yes) * P(windy \mid yes) * P(yes)}{P(sunny) * P(hot) * P(high) * P(windy)}$$

$$P(no \mid sunny \cap hot \cap high \cap windy) = \frac{P(sunny \mid no) * P(hot \mid no) * P(high \mid no) * P(windy \mid no) * P(no)}{P(sunny) * P(hot) * P(high) * P(windy)}$$

Types of Naïve Bayes

- Gaussian: It is used in classification and it assumes that features follow a normal distribution.

- Multinomial: It is used for discrete counts. For example, let’s say, we have a text classification problem. Here we can consider Bernoulli trials which is one step further and instead of “word occurring in the document”, we have “count how often word occurs in the document”, you can think of it as “number of times outcome number x_i is observed over the n trials”.

- Bernoulli: The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with ‘bag of words’ model where the 1s & 0s are “word occurs in the document” and “word does not occur in the document” respectively.

Application of Naïve Bayes

- Real time Prediction: Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- Multi class Prediction: This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- Recommendation System: Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

Pros and Cons

Pros:

- It is easy and fast to predict the class of the test dataset. It also performs well in multi-class prediction.
- Naïve Bayes easily scalable hence this algorithm use widely in the industry. And it’s a popular choice for text classification problems.

- When the assumption of independence holds, a Naïve Bayes classifier performs better compared to other models like logistic regression and you need less training data.
- It performs well in case of categorical input variables compared to numerical variables. For a numerical variable, normal distribution is assumed.

Cons:

- If a categorical variable has a category, which was not observed in training data set, then the model will assign a zero probability and will be unable to male predication. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.

- Another limitation of Naïve Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Implementation of Naïve Bayes

Titanic sank with over thousand passenger and crew still on board. Almost all those who jumped or fell into the water drowned or died within minutes due to the effects of cold, shock and incapacitation. However many factors affected survival rates that night, including one's location within the ship whether they were male or female whether they are children or grownups etc.

The general objective of this study is to build a data mining model which can produce a predictive report on survival chances of the passengers using some of the above mentioned factors into accounts. For my prediction I have chosen Naïve Bayes algorithm.

I start our exploratory data analysis with the data set provided on Kaggle. There are many attribute or features in train database but to get great predication result I have chosen Age, Passenger Class and Gender. In this data set there are many NAN value. So first I removed all NAN value with min value of all the data.

Then I calculate probability of Passenger class, probability of Gender and probability of Age with whole dataset. After that I took probability of how many people survived. Following that I calculated Conditional probabilities of class/survived, class/died, sex/survived, sex/died, Age/survived and Age/died.

Naive_Bayes.ipynb | test.csv | result.csv

```python
Class_counts=df_train['Pclass'].value_counts()
p_Class=Class_counts/len(df_train)

Sex_counts=df_train['Sex'].value_counts()
p_Sex=Sex_counts/len(df_train)

Age_counts=df_train['Age'].value_counts()
p_Age=Age_counts/len(df_train)

# Survival and Death probabilities
y_counts=df_train['Survived'].value_counts()
p_y=y_counts/len(df_train)

df_survived=df_train.loc[df_train['Survived'] == 1]
df_died=df_train.loc[df_train['Survived'] == 0]

# Conditional probabilities
#class/survived
class_survived_counts=df_survived['Pclass'].value_counts()
p_class_survived=class_survived_counts/len(df_survived)

# class/died
class_died_counts=df_died['Pclass'].value_counts()
p_class_died=class_died_counts/len(df_died)

#sex/survived
sex_survived_counts=df_survived['Sex'].value_counts()
p_sex_survived=sex_survived_counts/len(df_survived)

sex_died_counts=df_died['Sex'].value_counts()
p_sex_died=sex_died_counts/len(df_died)

#Age/survived
age_survived_counts=df_survived['Age'].value_counts()
p_age_survived=age_survived_counts/len(df_survived)

age_died_counts=df_died['Age'].value_counts()
p_age_died=age_died_counts/len(df_died)
```

And then I predict person is lived or not using Naïve Bayes equation.

Naive_Bayes.ipynb | test.csv | result.csv | Python 3

```python
age_died_counts=df_died['Age'].value_counts()
p_age_died=age_died_counts/len(df_died)
```

```python
[4]: def Bayes(py, px1y, px2y, px3y, px1, px2, px3):
    numerator=px1y*px2y*px3y*py
    denominator=px1*px2*px3
    p=numerator/denominator
    return p
```

```python
[5]: result_array=[]

for i in range(0,418):
    feature_class=df_test.iloc[i]['Pclass']
    feature_sex=df_test.iloc[i]['Sex']
    feature_age=df_test.iloc[i]['Age']

    P_Y1=Bayes(p_y[1], p_class_survived[feature_class], p_sex_survived[feature_sex], p_age_survived[feature_age], p_Class[feature_class], p_Sex[feature_sex], p_Age[fe
    P_Y0=Bayes(p_y[0], p_class_died[feature_class], p_sex_died[feature_sex], p_age_died[feature_age], p_Class[feature_class], p_Sex[feature_sex], p_Age[feature_age])

    if P_Y0 > P_Y1:
        result=0
    else:
        result=1

    result_array.append(result)


output = pd.DataFrame({'PassengerId': df_test.PassengerId,'Survived': result_array})
output.to_csv('result.csv', index=False)
```

After executing code I get output as result.csv file. In that, each passenger got passenger ID and that person is survived or not. If person is survived then 1 will show and if passenger is died than 0 will show. I check accuracy of my result.csv file and I got 0.76.

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
result.csv	just now	0 seconds	0 seconds	0.76794

Complete

Jump to your position on the leaderboard ▾

Reference

- https://www.wikipedia.org/
- https://www.kaggle.com/xopxesalmon/naive-bayes-classifier-from-scratch/data
- https://courses.analyticsvidhya.com/courses/take/naive-bayes/texts/11256768-key-terms-and-definitions
- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.iloc.html
- https://seaborn.pydata.org/examples/grouped_violinplots.html
- https://www.geeksforgeeks.org/python-pandas-dataframe-astype/

hochschule hof