Лабораторная работа 1

**Naive Bayes Classification   
(Наивная байесовская классификация)**

Цель работы

приобретение практических навыков в реализации наивного байесовского алгоритма.

**Теоретические сведения**

**Наивный байесовский алгоритм**

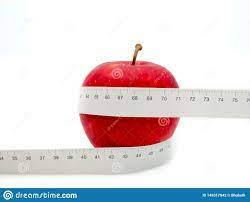
**Наивный байесовский алгоритм (НБА)** – это алгоритм классификации, основанный на [теореме Байеса](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%91%D0%B0%D0%B9%D0%B5%D1%81%D0%B0) с допущением о независимости признаков.

Другими словами, НБА предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака.

Пример:

Фрукт может считаться яблоком, если он

* красный,
* круглый и
* его диаметр составляет порядка 8 сантиметров.

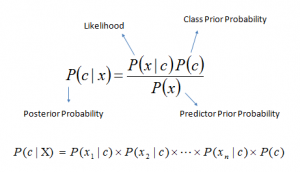


Даже если эти признаки зависят друг от друга или от других признаков, в любом случае они вносят независимый вклад в вероятность того, что этот фрукт является яблоком. В связи с таким допущением алгоритм называется «**наивным**».

Модели на основе НБА достаточно просты и крайне полезны при работе с очень большими наборами данных. При своей простоте НБА способен превзойти даже некоторые сложные алгоритмы классификации.

**Теорема Байеса**

Теорема Байеса позволяет рассчитать апостериорную вероятность *P(c|x)* на основе *P(c)*, *P(x)* и *P(x|c)*.



**P(c|x)** – апостериорная вероятность данного класса c (т.е. данного значения целевой переменной) при данном значении признака x.

**P(c)** – априорная вероятность данного класса.

**P(x|c)** – правдоподобие, т.е. вероятность данного значения признака при данном классе.

**P(x)** – априорная вероятность данного значения признака.

**Допущения наивного байесовского алгоритма**

Основным допущением наивного байесовского алгоритма является то, что каждая характеристика вносит **независимый** и **равный** вклад в конечный результат.

Сильные и слабые стороны наивного Байеса

**Основные сильные стороны:**

* Простой и быстрый способ прогнозирования классов, как в задачах классификации бинарных, так и мультиклассовых классов.
* В случаях, когда допущение независимости соответствует, алгоритм работает лучше по сравнению с другими классификационными моделями, даже с меньшим количеством обучающих данных.
* Разъединение распределений условных признаков класса означает, что каждое распределение может быть независимо оценено как одномерное распределение. Это помогает в решении проблем, связанных с проклятием размерности, и повышает производительность.

Основные недостатки **использования этого метода** являются:

* Хотя наивные байесовские алгоритмы довольно хорошие классификаторы, они, как известно, являются плохими оценщиками. Таким образом, вероятность того, что результаты будут получены, не следует воспринимать очень серьезно.
* Наивное предположение о независимости вряд ли будет соответствовать реальным данным.
* Когда набор тестовых данных имеет особенность, которая не наблюдалась в обучающем наборе, модель присвоит ему вероятность 0 и будет бесполезна для прогнозирования. Одним из основных способов избежать этого является метод сглаживания, являющийся оценкой Лапласа.

**Пример**

Ниже представлен обучающий набор данных, содержащий один признак **«Погодные условия»** **(weather)** и целевую переменную **«Игра» (play**), которая обозначает возможность проведения матча.

На основе погодных условий мы должны определить, состоится ли матч.

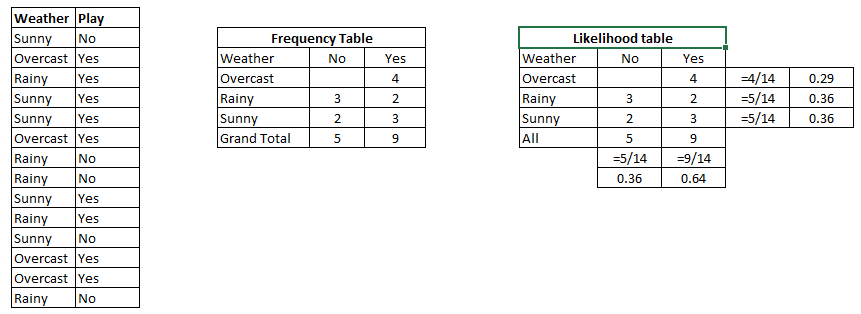
Чтобы сделать это, необходимо выполнить следующие шаги.

Шаг 1. Преобразуем набор данных в частотную таблицу (frequency table).

Шаг 2. Создадим таблицу правдоподобия (likelihood table), рассчитав соответствующие вероятности.

Например, вероятность облачной погоды (overcast) составляет 0,29, а вероятность того, что матч состоится (yes) – 0,64.

Шаг 3. С помощью теоремы Байеса рассчитаем апостериорную вероятность для каждого класса при данных погодных условиях. Класс с наибольшей апостериорной вероятностью будет результатом прогноза.

[](http://datareview.info/wp-content/uploads/2015/09/Bayes_41.png)

Sunny – Солнечная погода  
Rainy – Дождливая погода  
Overcast – Облачная погода

**Задача.** Состоится ли матч при солнечной погоде (sunny)?

P(Yes | Sunny) = P(Sunny | Yes) \* P(Yes) / P(Sunny)

P(Sunny | Yes) = 3 / 9 = 0,33

P(Sunny) = 5 / 14 = 0,36

P(Yes) = 9 / 14 = 0,64

Теперь рассчитаем P(Yes | Sunny):

P(Yes | Sunny) = 0,33 \* 0,64 / 0,36 = 0,60

Значит, при солнечной погоде более вероятно, что матч состоится.

Изучите еще примеры на странице <https://www.machinelearningmastery.ru/naive-bayes-intuition-and-implementation-ac328f9c9718/>

**Введение в Python**

Просмотрите оба видеофайла в папочке Python.

* 1. Python в Data Science.mp4 – фрагмент семинара по анализу данных. Одновременно с клипом выполнять рассматриваемое в клипе задание (файл - Введение в Python.ipynb).
  2. Intro to Data Analysis Visualization with Python, Matplotlib and Pandas Matplotlib Tutorial.mp4 - несложный английский язык. Этот клип можно смотреть и на сайте <https://www.youtube.com/watch?v=a9UrKTVEeZA>

3. Изучите также информацию, представленную в файле Naive\_Bayes.ipynb

Полезные библиотеки:

* **Numpy -** библиотека Python, которая позволяет обрабатывать массивы и эффективно применять к ним различные алгоритмы. Установить **Numpy** и изучить ее подробнееможно [здесь](https://numpy.org/)**:** <https://numpy.org/> .
* **SciKitLearn -** библиотека Python, которая представляет собой набор инструментов для большинства типов моделей машинного обучения. Узнать подробную информацию об этой библиотеке и установить ее можно здесь: <https://scikit-learn.org/stable/> .
* [**Seaborn**](https://seaborn.pydata.org/) - A python visualization library based on Matplotlib. Изучить и установить [**Seaborn**](https://seaborn.pydata.org/) можно [здесь](https://github.com/voila-dashboards/voila): <https://seaborn.pydata.org/> .
* **Pandas -** библиотека Python, которая позволяет загружать, очищать и обрабатывать данные. Для этих целей подойдут и такие альтернативы, как SQL, но Pandas намного проще и удобнее. Установить библиотеку и узнать о ней подробнее можно [здесь](https://pandas.pydata.org/): <https://pandas.pydata.org/> .
* [**Matplotlib**](https://matplotlib.org/) - A Python 2D plotting library. Узнать подробную информацию об этой библиотеке и установить ее можно здесь: <https://matplotlib.org/> .

**Пример на Python**

#Import Library of Gaussian Naive Bayes model

from sklearn.naive\_bayes import GaussianNB

import numpy as np

#assigning predictor and target variables

x= np.array([[-3,7],[1,5], [1,2], [-2,0], [2,3], [-4,0], [-1,1], [1,1], [-2,2], [2,7], [-4,1], [-2,7]])

Y = np.array([3, 3, 3, 3, 4, 3, 3, 4, 3, 4, 4, 4])

#Create a Gaussian Classifier

model = GaussianNB()

# Train the model using the training sets

model.fit(x, Y)

#Predict Output

predicted= model.predict([[1,2],[3,4]])

print predicted

Output: ([3,4])

Еще вариант

import pandas as pd

»’  
P(A)\*P(B|A)  
P(A|B) = ————-  
SUM\_i:P(Ai)\*P(B|Ai)  
»’

def prior(data, column\_a, value\_a):  
return len(data[data[column\_a] == value\_a]) / (len(data)+0.00001)

def likelyhood(data\_only\_a, column\_b, value\_b):  
return len(data\_only\_a[data\_only\_a[column\_b] == value\_b]) / (len(data\_only\_a)+0.00001)

def evidence(data, column\_a, column\_b, value\_b):  
P\_B = 0  
for a\_i in data[column\_a].unique():  
P\_B += prior(data, column\_a, a\_i) \  
\* likelyhood(data[data[column\_a] == a\_i], column\_b, value\_b)  
return P\_B

column\_a = ‘Play’  
column\_b = ‘Weather’  
data = pd.DataFrame({column\_a:[‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘no’, ‘no’,’yes’,’yes’,’yes’, ‘yes’,’yes’,’yes’,’yes’],  
column\_b:[‘Rainy’, ‘Rainy’, ‘Rainy’, ‘Rainy’, ‘Rainy’, ‘Sunny’, ‘Sunny’,’Sunny’,’Sunny’,’Sunny’, ‘Overcast’,’Overcast’,’Overcast’,’Overcast’]})

B\_hypothes = ‘Sunny’  
A\_value = ‘yes’  
\_prior = prior(data, column\_a, A\_value)  
\_likelyhood = likelyhood(data[data[column\_a] == A\_value], column\_b, B\_hypothes)  
\_evidence = evidence(data, column\_a, column\_b, B\_hypothes)

#P(A\_value|B\_hypothes)  
P = \_prior\*\_likelyhood/\_evidence  
print(f’P({A\_value}|{B\_hypothes}) =’, P)

## **Советы по оптимизации модели**

* Если значения непрерывных признаков не обладают нормальным распределением, тогда с помощью соответствующего преобразования необходимо привести их к такому распределению.
* Если тестовый набор данных имеет проблему «нулевой частоты, необходимо применить сглаживание по Лапласу.
* Если два признака имеют высокую корреляцию, один из них следует удалить, иначе они будут вносить удвоенный вклад, завышая тем самым свою значимость.
* Наивный байесовский классификатор имеет набор параметров, доступных для настройки. Например, *alpha=1* – активирует сглаживание по Лапласу, *fit\_prior=[True | False]*– определяет, обучать ли модель априорным вероятностям классов. Полный список параметров можно найти [здесь](http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB):

<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB> .

Рекомендуется сосредоточиться на предобработке данных и отборе признаков.

* Следует отметить, что в случае НБА использование ансамблевых методов, таких как бэггинг (bagging) и бустинг (boosting), не дает результатов. Данные подходы направлены на уменьшение дисперсии, что неприменимо по отношению к НБА.

Задание

1. Определиться с областью(ями) данных, которая(ые) вам интересна(ы), и выбрать 2 набора данных (dataset) для исследования.

## [Kaggle](https://www.kaggle.com/) (<https://www.kaggle.com>)

На Kaggle представлено более 50 000 датасетов в широчайшем диапазоне тематик — от здравоохранения до мультфильмов. Датасеты Kaggle, используемые в соревнованиях, часто более детализированы, чем общедоступные датасеты.

## [Google Cloud Public Datasets](https://cloud.google.com/solutions/datasets) (<https://cloud.google.com/solutions/datasets> )

Google Сloud содержит более сотни датасетов, размещенных в BigQuery и облачном хранилище. Датасеты взяты из различных источников, таких как GitHub, Бюро переписи населения США, NASA, BitCoin и многих других.

## [Amazon Web Services Open Data Registry](https://registry.opendata.aws/) (<https://registry.opendata.aws/> )

Amazon Web Services позволяет загрузить датасеты или исследовать их в Elastic Compute Cloud. Open Data Registry является частью программы AWS Public Dataset, направленной на демократизацию доступа к данным.

## [Data.gov](https://www.data.gov/) (<https://www.data.gov/> )

Data.gov является основным хранилищем открытых датасетов правительства США. Большинство датасетов находятся в открытом доступе, однако для остальных требуется получить разрешения для их загрузки. Представленные на сайте данные касаются климата, сельского хозяйства и энергетики.

## [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/index.php) (<https://archive.ics.uci.edu/ml/index.php> )

Старейший источник данных, работающий с 1987-го года. Датасеты UCI отлично подходят для машинного обучения благодаря возможности настройки параметров загрузки.

## [Global Health Observatory](https://www.who.int/data/gho) (https://www.who.int/data/gho)

Хранилище данных ВОЗ, содержащих информацию о различных инфекционных и неинфекционных заболеваниях, психическим расстройствам и лекарствам.

## [Earthdata](https://earthdata.nasa.gov/) (https://earthdata.nasa.gov/)

Датасеты NASA, содержащие информацию об атмосфере Земли, океанах, криосфере, солнечных вспышках. В Earthdata есть инструменты для обработки, категоризации, поиска и визуализации данных.

Просмотрите также следующие статьи:

Лучшие датасеты для машинного обучения и анализа данных

<https://tproger.ru/translations/the-best-datasets-for-machine-learning-and-data-science/>

Практика Data Science: где искать датасеты и что с ними делать — отвечают эксперты

<https://tproger.ru/experts/data-science-practice/>

Подборка датасетов для машинного обучения

<https://vc.ru/dev/68180-podborka-datasetov-dlya-mashinnogo-obucheniya>

52 датасета для тренировочных проектов

<https://habr.com/ru/company/edison/blog/480408/>

Открытые наборы данных Azure

<https://docs.microsoft.com/ru-ru/azure/open-datasets/dataset-catalog>

1. Для каждого выбранного датасета сформулировать не менее 2 задач и определить наиболее вероятное решение с помощью Python. Использовать графическую визуализацию решений.
2. Составить отчет по лабораторной работе. Электронную копию положить на гугл-диск в папку Студенты, создав в ней папку со своей фамилией.



**ДЕДЛАЙН – окончание лабораторной работы №2.**

