# **Лабораторная работа 3. Байесовский классификатор**

## **Цели**

Приобрести навыки построения байесовского классификатора

## **Задачи**

1. Построить модель гауссовского наивного байесовского классификатора для бинарной классификации
2. Оценить точность модели

## **Теоретические сведения**

Методические указания для решения поставленного задания

### **Бинарная классификация**

Наивный байесовский алгоритм – алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков. Другими словами, алгоритм предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака. Например, с помощью такой модели можно определить письмо, содержащее спам. В данной работе модель будет определять, поступит ли студент в магистратуру, на основе его дохода и размера апартаментов

В данной работе вы будете работать с гауссовским наивным байесовским классификатором. Модификация заключается в том, что классификатор предполагает гауссовское распределение объектов

Для того, чтобы рассчитать модель классификатора используются следующие формулы, где – число объектов класса

Мультиколлинеарность – тесная корреляционная взаимосвязь между факторами. Мультиколлинеарность может стать причиной переобучения модели, что приведет к неверным результатам её работы. Кроме того, избыточные параметры увеличивают сложность модели, а значит и время ее тренировки. Также мультиколлинеарность факторов плоха тем, что модель будет содержать избыточные переменные, а это значит, что:

1. Осложняется интерпретация параметров как величин действия факторов, параметры теряют смысл и следует рассматривать другие переменные;
2. Оценки параметров ненадежны – получаются большие стандартные ошибки, которые меняются с изменением объема наблюдений, что делает модель непригодной для прогнозирования.

Для оценки мультиколлинеарности используется матрица парных коэффициентов корреляции

### **Построение модели с помощью scikit-learn**

Пример построения модели будет приведён ниже, демонстрация будет проводиться на данных из набора данных [Go to college dataset](https://www.kaggle.com/datasets/saddamazyazy/go-to-college-dataset) (рус. Набор данных о поступлении в колледж)

#### **Подключение библиотек**

Подключение библиотек

**import** pandas **as** pd  
**import** seaborn **as** sns  
  
**from** sklearn **import** metrics  
**from** sklearn.model\_selection **import** train\_test\_split  
**from** sklearn.naive\_bayes **import** GaussianNB

#### **Работа с набором данных**

Подключение набора данных

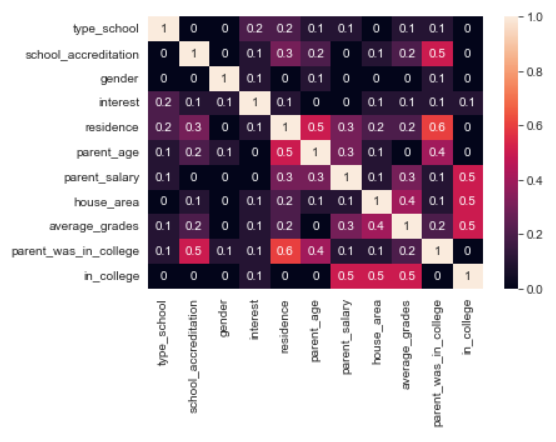
PATH = "college.csv"  
dataset = pd.read\_csv(PATH)

Факторизация значений в столбцах, содержащих строковые значения

**for** column **in** dataset:  
 **if** type(dataset[column][0]) **is** str:  
 dataset[column] = pd.factorize(dataset[column])[0]

Быстрая, визуальная проверка данных на мультиколлинеарность – не выявила особо сильных связей

sns.heatmap(  
 round(  
 abs(dataset.corr()),  
 1,  
 ),  
 annot=True,  
)



#### **Построение модели**

Разделение выборки на тренировочную и тестовую часть, а также на параметры и эталоны

train\_input, test\_input, train\_output, test\_output = train\_test\_split(  
 dataset.drop('in\_college', axis=1),  
 dataset["in\_college"],  
 test\_size=0.2  
)

Обучение, тестирование и проверка модели

model = GaussianNB()  
model.fit(train\_input, train\_output)  
  
predictions = model.predict(test\_input)  
accuracy = metrics.accuracy\_score(predictions, test\_output)  
  
print(f"Точность модели на тестовом участке = {accuracy}")

Точность модели на тестовой части набора данных составила 0.735

## **Задание**

Выбрать с сайта [kaggle.com](https://www.kaggle.com/) набор данных в формате .csv, пригодный для построения бинарной классификации, загрузить и подготовить его к дальнейшей обработке. Наборы данных не должны повторяться внутри группы. Задание индивидуальное. Требования:

1. Построить модель гауссовского наивного байесовского классификатора
2. Оценить точность модели
3. Указать какие знания можно получить из набора
4. Сохранить IPython Notebook

### **Продвинутое задание**

Построить вторую модель, без использования средств scikit-learn