# Московский авиационный институт (национальный исследовательский университет)

## Факультет информационных технологий и прикладной математики

## Кафедра вычислительной математики и программирования


**Лабораторная работа №1 по курсу «Машинное обучение»**


Студент: К. О. Вахрамян
Преподаватель: Ахмед Самир Халид
Группа: М8О-306Б

Дата:
Оценка:
Подпись:

**Москва, 2021**

# Лабораторная работа №1

**Задача:**

Найти себе набор данных (датасет), для следующей лабораторной работы, и проанализировать его. Выявить проблемы набора данных, устранить их. Визуализировать зависимости, показать распределения некоторых признаков. Реализовать алгоритмы К ближайших соседа с использованием весов и Наивный Байесовский классификатор и сравнить с реализацией библиотеки sklearn.

# 1 Описание

Для анализа и классификации я взял датасет с информацие о годовом доходе физических лиц. Квалификация здесь бинарная: нужно предсказать, доход больше $50K или нет.

Первой задачей было подготовить данные. Я избалвился от пропусков, убрал неинформативные столбцы. Далее я преобразовал данные для применения к ним моделей.

## Алгоритм K ближайших соседей

Идея данного алгоритма состоит в следующем - пусть у нас есть некоторая обучающая выборка $X = ((x_1, y_1), ..., (x_n, y_n))$ где $x_i$ - набор признаков для конкретного элемента, $y_i$ - то, к какому классу данный элемент относиться. Зададим некоторую функцию $\varrho(x_i, y_i)$, которая будет как-то адекватно показывать насколько элементы $x_i$ и $y_i$ похожи друг на друга(или близки друг к другу). Затем для каждого элемента, который необходимо классифицировать, посчитаем расстояния до всех элементов обучающей выборки и выберем из них $k$ ближайших соседей. Для того чтобы избежать неопределенности при классификации каждому соседу сопоставим некоторый вес, который будет определяться весовой функцией, зависящей от расстояния между классифицируемым элементом и соседом. Я использовал функцию равную обратному квадрату расстояния. Затем для каждого класса суммируем получившиеся веса и относим классифицируемый элемент к тому классу, сумма весов которого получилась наибольшей. Стоит отметить, что для данного алгоритма иногда стоит проводить предобработку датасета, поскольку признаки могут быть распределены в разных промежутках, и, соответственно, тот, что имеет большее значение будет вносить больший импакт в функцию расстояния. В моем случае это не было необходимо, поскольку значения всех признаков находятся в промежутке от 1 до 5.

Для реализации данного мной был написан класс SimpleKNNClassifier с интерфейсом, похожим на интерфейс объектов из библиотеки sklearn, а именной функцией fit для обучения, которой на вход передаются данные обучающей выборки и функцией predict, которой на вход передаются признаки классифицируемого элемента. В функции fit ничего интересного не происходит - полученные данные просто кладутся в поля класса, а в функции predict реализован описанный выше алгоритм.

## Наивный байесовский классификатор

Данный метод основан на использовании форумлы Байеса:

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

Считаем выборочные средние и дисперсии для каждого признака в зависимости от класса. Для каждого класса находим оценку вероятности того, что случайное наблюдение принадлежит данному классу путём деления количества наблюдений с данным классом на общее число наблюдений. Находим условную вероятность признаком при условии данного класса с помощью плотности вероятности нормального распределения. Предсказываем класс, для которого вероятность по формулей Байеса наибольшая.

# 2 Результаты

```
Max accuracy with my knn 0.8344216934144991
Max accuracy with skl knn 0.8409518539014942

Accuracy of custom Naive Bayes: 0.8057553956834532
Accuracy of sklearn Naive Bayes: 0.7967349197565025
```

# 3 Выводы

В ходе выполнения работы был подготовлен и проанализирован датасет. Также реализован алгоритм knn с точностью, незначительно меньшей алгоритма из библиотеки sklearn, точность Наивного Байэсовского классификатора и вовсе оказалась выше библиотченого.