Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет: “Информационные технологии и прикладная математика”
Кафедра: 806 “Вычислительная математика и программирование”

Лабораторная работа № MO1

по курсу “Искусственный интеллект”

Студент	Полей-Добронравова А.В.
Группа	М8О-307Б-18
Преподаватель	А. С. Халид
Дата	
Оценка	

Москва, 2021

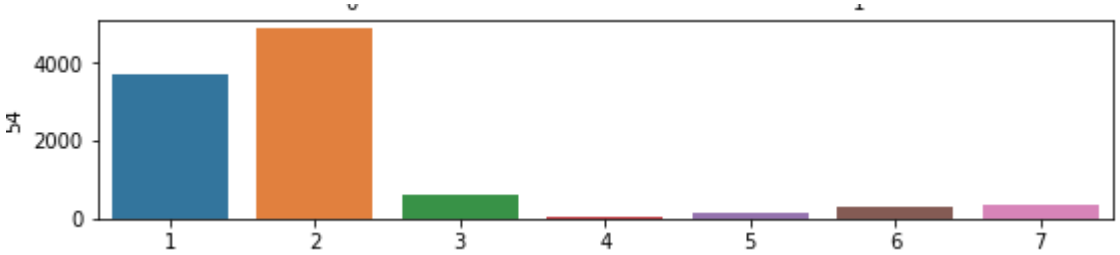
Постановка задачи

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

Датасет

Выбранный мной датасет я взяла в одном из домашних заданий по пройденному мной курсу по машинному обучению, он представляет собой датасет растений в лесу. Я загрузила его в начале на гугл диск, а потом через гугл колаб подгрузила к себе в рабочую директорию ноутбука.

Представлено 7 классов растений, и количество представителей каждого класса я отобразила последним графиком в исследовании признаков:



Классы несбалансированные, что плохо для машинного обучения, но ввиду того, что некоторые классы представлены совсем в малом количестве, например 4 класс, я не стала убирать из датасета элементы 1 и 2 класса, хотя можно было выкинуть половину, чтобы улучшить обучение.

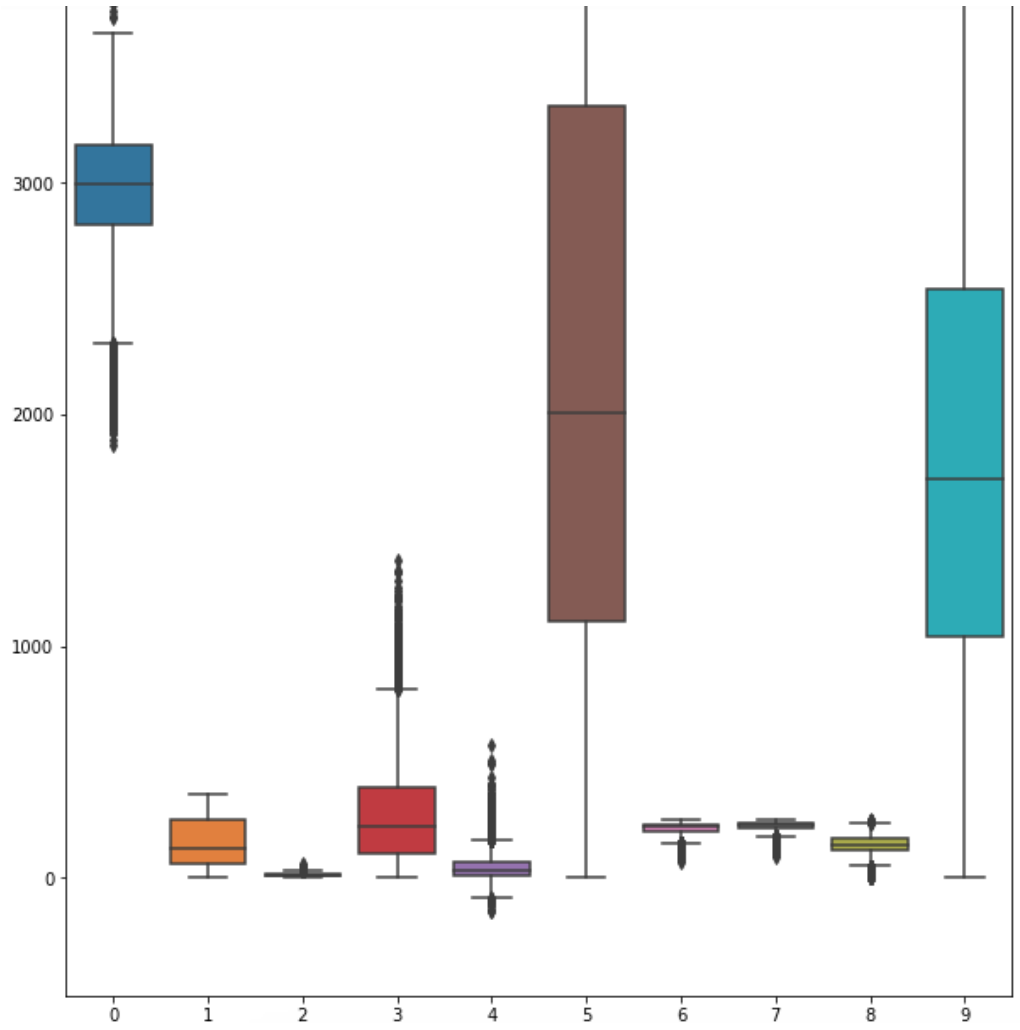
При анализе датасета я решила обнаружить и обнулить пропущенные значения(NaN), потому что из-за них часто многие функции не могут отработать корректно. Но моё исследование показало, что пропущенных значений в датасете нет:

```
   0  1  2  3  4  5  6  7  8  9 10 11 12  ... 42 43 44 45 46 47 48 49 50 51 52 53 54
0  0  0  0  0  0  0  0  0  0  0  0  0  0  ...  0  0  0  0  0  0  0  0  0  0  0  0  0

[1 rows x 55 columns]
```

В представленной выше таблице должно было отобразиться количество NaN в каждом из столбцов признаков.

Далее я нарисовала графики распределения всех признаков. Для числовых признаков распределение следующее:



Для категориальных признаков я нарисовала гистограммы, на которых видно, что некоторые признаки представлены только в одном значении, например 28 признак. Но я не стала его удалять, т.к. его анализ занимает не так много времени для анализаторов.

Реализация

Реализация КНН представлена в функции `def classifyKNN (trainData, testData, k, numberOfClasses),` в которую передаются данные для обучения, для тестирования, число соседей для учитывания k, число классов. Моя функция работает только для моего датасета, но для того, чтобы ее можно было использовать для любых датафреймов, нужно только изменить в функции расстояния dist крайнее значение для цикла for на число колонок.

Для того, чтобы продемонстрировать работу КНН, я урезала датасет до 2100 строк в обучающей выборке, что составляет 70% от всего набора. Сделано это для того, чтобы прохождение длилось быстрее, и на таких

данных предсказания я получила через 8 минут. Получена точность 68% при количестве соседей 3:

```
[ ]  testData = test
     print('RESULTS:')
     sum_ = 0
     j = 0
     for i, row in testData.iterrows():
       sum_ += int((testDataLabels[j]+1)==row[54])
       j += 1
     sum_ /= float(len(testData))
     print('Accuracy: ', sum_ )

     RESULTS:
     Accuracy:  0.6844444444444444
```

Точность понижается при увеличении количества соседей, поэтому я остановилась на 3 соседях.

На удивление точность оказалась выше, чем точность у библиотечного КНН из sklearn, в нём точность 0,66. Связано это скорее с тем, что для вызова библиотечной функции я разделила датасет заново, и получились другие наборы обучающей и тестовой выборки.

Байесовский классификатор реализован в функциях btrain и bclassify. Изначальный код я взяла из статьи на Хабр (https://habr.com/ru/post/120194/), но мне пришлось его модифицировать для того, чтобы он запускался на датафреймах pandas. Полученная точность 48%, что очень мало. Я считаю, что проблема в том, что я урезала датасет, и Байесовскому классификатору не хватило данных для хорошего обучения. У библиотечной реализации точность вышла совсем плохая: 39%. Но ввиду малого количества часов для решения лабораторной, я не успела протестировать Байесовские классификаторы на не урезанной версии датасета.

Вывод

Основные проблемы подготовки данных перед машинным обучением были отображены: несбалансированность классов, маленькое количество данных для обучения. Также я реализовала простейшие модели МО: КНН и Байесовский классификатор. Точность у КНН была выше на 10%.