# Московский авиационный институт (национальный исследовательский университет)

# Институт информационных технологий и прикладной математики

# Кафедра вычислительной математики и программирования

# Лабораторная работа №1 по курсу «Искусственный интеллект»

# Студентка: Кочуйкова Д.В. Группа: М8О-307Б-17

Москва, 2020Постановка задачи:

Необходимо реализовать алгоритмы машинного обучения. Применить данные алгоритмы на наборы данных, подготовленных в первой лабораторной работе. Провести анализ полученных моделей, вычислить метрики классификатора. Произвести тюнинг параметров в случае необходимости. Сравнить полученные результаты с моделями реализованными в scikit-learn. Аналогично построить метрики классификации. Показать, что полученные модели не переобучились. Также необходимо сделать выводы о применимости данных моделей к вашей задаче.

* Логистическая регрессия
* KNN
* SVM
* Дерево Решений

Реализация

Логистическая Регрессия

Описание модели

Для мультиклассовой классификации, для большого класса распределений апостериорные вероятности задаются преобразованием softmax линейных функций переменных признаков, так что:

Там я использовал максимальное правдоподобие для отдельного определения условных плотностей классов и априорных значений класса, а затем нашел соответствующие апостериорные вероятности с помощью теоремы Байеса, тем самым неявно определяя параметры {wk}. Использование максимальной вероятности для определения параметров этой модели напрямую. Для этого потребуются производные от yk по всем активациям a\_j. Это дало:

,

где - элементы единичной матрицы. Затем нужно записать функцию правдоподобия. Это легче всего сделать, используя схему кодирования 1 - из - K, в которой целевой вектор для вектора признаков , принадлежащего классу , представляет собой двоичный вектор со всеми элементами, равными нулю, кроме элемента k, который равен единице. Функция правдоподобия тогда выглядит:

где , а T - матрица целевых переменных размером N × K с элементами . Применяя логарифм, получается

То есть функция кросс-энтропийной ошибки для задачи мультиклассовой классификации. Градиент функции ошибки относительно одного из векторов параметров :

,

Для комбинации функции активации логистической сигмоиды и функции кросс-энтропийной ошибки, а также для функции активации softmax с мультиклассовой функцией кросс-энтропийной ошибки получается данная форма. С помощью метода Ньютона-Рафсона, получаю алгоритм IRLS для задачи мультикласса. Это требует оценки матрицы Гессе, которая содержит блоки размера M × M, в которых блок j, k задается как

.

**Результат работы**

Ошибки на кросс валидации:

Accuracy = 0.58

Precision = 0.58

Roc auc = 0.5059324776183799

Ошибки на выборках

Train Test

Accuracy = 0.6537313432835821 | 0.6181818181818182

Precision = 0.6537313432835821 | 0.6181818181818182

Roc auc = 0.5 | 0.5

**Логистическая регрессия с Sklearn.**

Ошибка на выборках:

Accuracy = 0.706060606060606

Ошибки на кросс валидации:

Accuracy = 0.7212121212121212

Ошибки на выборках

Train Test

Accuracy = 0.7388059701492538 | 0.7212121212121212

Precision = 0.7388059701492538 | 0.7212121212121212

Roc auc = 0.6785939222169737 | 0.6758870214752568

**Выводы по модели:**

1. Из того, что метрики классификации на тренировочной и тестовой выборках почти не отличаются следует, что модель не переобучилась.

2. Модель логистической регрессии показалась довольно сложной и заняла достаточно времени. Также, из полученных оценок можно сделать вывод, что данная модель не подходит для решения задачи мультиклассовой классификации.

KNN

Описание модели

Для классификации каждого из объектов тестовой выборки необходимо последовательно выполнить следующие операции:

* Вычислить расстояние до каждого из объектов обучающей выборки
* Отобрать k объектов обучающей выборки, расстояние до которых минимально
* Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей

Для простоты я выбрал двумерное пространство, в котором случайным образом на участке от 0 до 5 по каждой из осей выбирается местоположение мат.ожидания двумерного гауссиана со среднеквадратичным отклонением 0.5. Значение 0.5 выбрано, чтобы объекты оказались достаточно хорошо разделимыми (следует из правила трех сигм). Для определения расстояния между объектами можно использовать не только евклидово расстояние: также применяются манхэттенское расстояние, косинусная мера, критерий корелляции Пирсона и др.

Основная функция алгоритма. На вход приходит матрица расстояний между объектами обучающей и тестовой выборки, метки обучающей выборки, число ближайших «соседей». На выходе предсказанные метки для новых объектов и вероятности каждой метки.

Результат выполнения

**Реализованная модель**

CV scores: [0.6194029850746269, 0.6343283582089553, 0.7388059701492538, 0.6567164179104478, 0.5895522388059702]

Train data accuracy: 0.6477611940298507

Ошибки на кросс валидации:

Accuracy = 0.6393939393939394

Precision = 0.6393939393939394

Roc auc = 0.5793650793650793

Ошибки на выборках

Train Test

Accuracy = 0.7313432835820896 | 0.6393939393939394

Precision = 0.7313432835820896 | 0.6393939393939394

Roc auc = 0.6698452999527633 | 0.5793650793650793

**KNN с Sklearn**

Ошибки на выборках

Train Test

Accuracy = 0.7373134328358208 | 0.6515151515151515

Precision = 0.7373134328358208 | 0.6515151515151515

Roc auc = 0.6744115099984255 | 0.5997899159663865

**Вывод по модели:**

1. Из того, что метрики классификации на тренировочной и тестовой выборках почти не отличаются следует, что модель не переобучилась.

2. Модель KNN считается не сложной, но даже при этом на тренировочных и тестовых данных дает результат не хуже логрега.

SVM

Описание модели

Мультиклассовые SVM классифицируют входной вектор x ∈ в один из k классов, используя следующее простое правило:

**,**

Каждый вектор ∈ можно рассматривать как прототип, представляющий m-й класс, а - как оценку m-го класса по x. Следовательно, уравнение выбирает класс с наивысшим баллом. Для n обучающих примеров ∈ и связанных с ними меток ∈ [k] мультиклассовая SVM-формулировка оценивает оценки , решив следующую задачу оптимизации:

,

где C > 0 - параметр регуляризации и = 0, если u <0, и u в противном случае. Это уравнение означает, что для каждого учебного случая мы не несем потерь, если оценка правильного класса больше, чем оценка «ближайшего» класса хотя бы на 1. Оценками решив следующую задачу оптимизации:

,

с учетом , если ,

.

Градиент f играет важную роль и определяется:

.

В таком случае, оптимальное решение будет выглядеть:

.

Чем больше , тем больше . Остановка происходит, если .

Результат выполнения

Ошибки на кросс валидации:

Accuracy = 0.5509999999999999

Precision = 0.5509999999999999

Roc auc = 0.553014903114113

Ошибки на выборках

Train Test

Accuracy = 0.6537313432835821 | 0.6181818181818182

Precision = 0.6537313432835821 | 0.6181818181818182

Roc auc = 0.5 | 0.5

**SVM с Sklearn**

Ошибки на выборках

Train Test

Accuracy = 0.6537313432835821 | 0.6181818181818182

Precision = 0.6537313432835821 | 0.6181818181818182

Roc auc = 0.5 | 0.5

**Вывод по модели:**

1. Из того, что метрики классификации на тренировочной и тестовой выборках почти не отличаются следует, что модель не переобучилась.

2. Реализованная модель показала результаты хуже чем предыдущие модели, что говорит о ее неприменимости к решению этой задачи

**Решающее дерево**

Описание модели

**Дерево решений представляет собой двоичное дерево:**

* Узел — это одна входная переменная (x) и точка ее разделения.
* Конечные узлы (терминальные) дерева содержат выходную переменную (y), которая используется для прогнозирования.

**Рекурсивный алгоритм**

1. На каждом шаге выбирается признак, при разделении по которому прирост информации оказывается наибольшим. Создание разделения включает в себя три части:
2. Вычисление показателя Джини:
3. Разделение набора данных:
   1. Проверка на то, находится ли значение атрибута ниже или выше значения разделения.
   2. Присвоение его левой или правой группе соответственно.
4. Оценка всех разделений:

* Проверяются значения каждого атрибута как разделение кандидатов, оценивается стоимость разделения и находится наилучшее возможное разделение, которое мы могли бы сделать.
* Как только будет найдено лучшее разделение, оно используется в качестве узла в дереве решений.

1. Выборка делится на левую и правую часть.

* Выбирается разделение с лучшей стоимостью.
* Все входные переменные и все возможные точки разделения оцениваются и выбираются наилучшие на основе функции стоимости.

1. Процедура повторяется рекурсивно к каждой из частей, пока энтропия не окажется равной нулю или очень малой величине.

Дополнительные условия останова:

* Достижение нужной глубины дерева.
* Достижение минимальной записи узла.

**Представление дерева**

* Используется словарь для представления узла в дереве решений, поскольку мы можем хранить данные по имени.
* При выборе наилучшего разделения и использовании его в качестве нового узла для дерева, хранится индекс выбранного атрибута, его значения, по которому нужно разделяться, и две группы данных, разбитые по выбранной точке разделения.
* Каждая группа данных представляет собой собственный небольшой набор данных, состоящий только из тех строк, которые были назначены левой или правой группой в процессе разделения.

**Предсказание**

После создания дерева можно перемещаться с новой строкой данных, следующей за каждой ветвью с разбиениями, пока не будет сделан окончательный прогноз.

Результат выполнения.

Ошибки на кросс валидации:

Accuracy = 0.651

Precision = 0.651

Roc auc = 0.5944598275488645

Ошибки на выборках

Train Test

Accuracy = 0.8283582089552238 | 0.6484848484848484

Precision = 0.8283582089552238 | 0.6484848484848484

Roc auc = 0.7866182490946307 | 0.6018907563025211

**Дерево решений с Sklearn**

Ошибки на выборках

Train Test

Accuracy = 0.8701492537313433 | 0.6121212121212121

Precision = 0.8701492537313433 | 0.6121212121212121

Roc auc = 0.8631809951188789 | 0.5830999066293183

**Вывод по модели:**

1. Из того, что метрики классификации на тренировочной и тестовой выборках отличаются, возможно наличие переобучения.

2. Как видно из полученных результатов дерево решений очень хорошо подходит для решения данной задачи классификации.