Московский авиационный институт
(Национальный исследовательский университет)

Институт: «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»
Дисциплина: «AI/DL»

Курсовой проект

Студент: Хренов Геннадий
Группа: 80-407Б
Преподаватель: Вишняков Б.В.
Дата:
Оценка:

Москва, 2021

1. Введение

В ходе выполнения работы мне предстоит более подробно познакомиться с задачей классификации, которую я решал в лабораторной работе №4, проанализировать свое решение и попытаться его улучшить, а также еще раз пройтись по всем основным этапам решения задач машинного обучения.

2. Постановка задачи

Задан датасет, содержащий признаки 3 разных классов вин. Необходимо создать модель, которая по набору признаков образца способна определять его класс, то есть решать задачу классификации. Качество работы модели оценивать метрикой accuracy.

3. Описание данных

Набор данных подготовлен Калифорнийским университетом в Ирвине. Этот набор данных является результатом химического анализа вин, выращенных в одном и том же регионе Италии, но полученных из трех разных сортов. Анализ определил количества 13 компонентов, обнаруженных в каждом из трех типов вин, а именно:

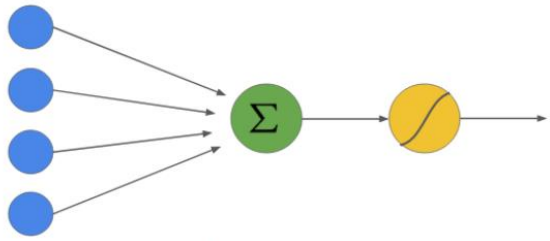
- Алкоголь
- Яблочная кислота
- Пепел
- Щелочность золы
- Магний
- Общие фенолы
- Флаваноиды
- Нефлаваноидные фенолы
- Проантоцианы
- Интенсивность цвета
- Оттенок
- OD280 / OD315 разбавленных вин
- Пролин

Каждый образец относиться к одному из 3 классов:

- 1 класс – 59 образцов
- 2 класс – 71 образцов
- 3 класс – 48 образцов

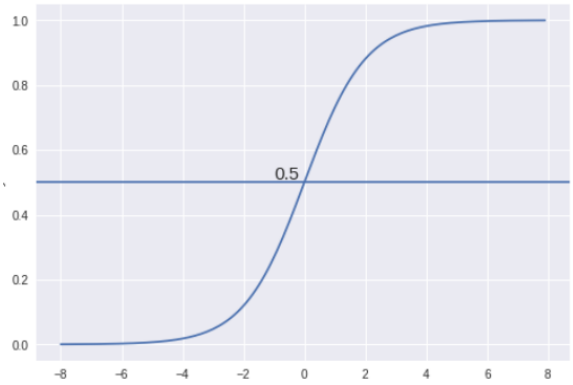
4. Теоретическая часть

Для решения задачи используем логистическую регрессию - фундаментальный метод классификации. В логистической регрессии взвешенная сумма входных данных пропускается через функцию, которая может отображать любое действительное значение в диапазоне от 0 до 1.



Logistic Regression

Используемая функция активации - сигмоида. Ее график выглядит так:



Значение сигмоидной функции всегда лежит между 0 и 1. Значение точно равно 0,5 при X = 0. Мы можем использовать 0,5 в качестве порога вероятности для определения классов при бинарной классификации.

Модель линейной регрессии может быть представлена уравнением.

$$h(x) = \theta^T x$$

Затем мы применяем сигмоидную функцию к выходу линейной регрессии

$$h(x) = \sigma(\theta^T x)$$

где сигмовидная функция представлена

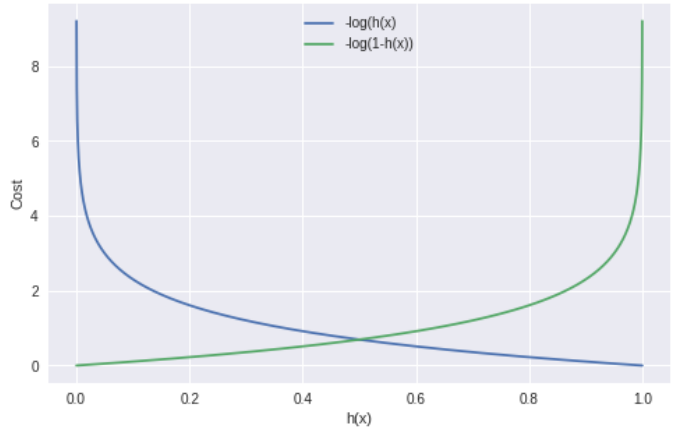
$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

Тогда гипотеза для логистической регрессии становится

$$h(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Функция стоимости для одного примера обучения может быть задана как:

$$cost = \begin{cases} -log(h(x), & \text{if } y = 1 \\ -log(1 - h(x)), & \text{if } y = 0 \end{cases}$$



Стоимость всех обучающих примеров можно рассчитать, взяв среднее по стоимости всех обучающих образцов

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^i log(h(x^i)) + (1 - y^i) log(1 - h(x^i))]$$

Логистическая регрессия подходит только для бинарной классификации, а у нас 3 класса. Поэтому для решения нашей задачи мы обучем несколько моделей по принципу "один против всех", а класс определяется как argmax выходов моделей.

Также я добавил регуляризацию L1. L1-регуляризация отбирает наиболее важные факторы, которые сильнее всего влияют на результат. Реализуется регуляризация путем добавления слогаемого после стоимости:

$$J_{LASSO} = -\sum_{i=1}^{N}(t_n log\, y_n + (1 - t_n)\, log(1 - y_n)) + \lambda \|w\|_1.$$

5. Предобработка данных

   Данные полностью заполнены, пропусков и выбросов нет. Выделяем номер класса, это первый столбец. Проверяем, что в каждом классе достаточное количество образцов. В таблице присутствуют значения как 0.26 так 1050 – необходимо масштабирование. Разбиваем данные на тестовое и обучающее множества. Также в качестве дополнения к лабораторной работе я решил уравнять количество образцов в каждом классе.

6. Реализация алгоритма

```
class_number = 3
l = 5
class myLogisticRegression:
  def sigmoid(self, x, weight):
    z = np.dot(x, weight)
    return 1 / (1 + np.exp(-z))

  def loss(self, h, y):
    return (-y * np.log(h) - (1 - y) * np.log(1 - h)).mean()
```

```
def gradient_descent(self, X, h, y, theta, l):
  thetaReg = theta[1:]
  reg = np.concatenate([np.array([0]), (l / len(X)) * (thetaReg)])
  return (np.dot(X.T, (h - y)) / len(X)) + reg

def fit(self, X, y, lr=0.1, iterations= 40):

  self.all_theta = np.zeros((class_number, X.shape[1]))

  acc = np.array([accuracy(LR.predict(X_train), y_train)]) #for grafs
  acc_test = np.array([accuracy(LR.predict(X_test), y_test)])

  for _ in range(iterations):
    for i in range(1,class_number+1):
      y_i = np.array([1 if label == i else 0 for label in y])
      sigma = self.sigmoid(X, self.all_theta[i-1,:])
      dW = self.gradient_descent(X , sigma, y_i, self.all_theta[i-1,:], l)
      self.all_theta[i-1,:] -= lr * dW

    acc = np.append(acc, accuracy(LR.predict(X_train), y_train))
    acc_test = np.append(acc_test, accuracy(LR.predict(X_test), y_test))

  plt.plot(acc)
  plt.plot(acc_test)
  plt.show()

def predict(self, X):
  h = self.sigmoid(X, self.all_theta.T)
  h_argmax = np.argmax(h, axis=1)
  return h_argmax+1
```

7. Метрика качества

В задачах бинарной классификации результаты могут быть следующие:

- истинно положительные (TP)
- истинно отрицательные (TN)
- ложно положительные (FP)
- ложно отрицательные (FN)

Исходя из цены ошибки первого или второго рода, а также важности получения истинного результата для конкретной задачи обычно выбирается одна из метрик: Accuracy, Precision, Recall.

Но в нашем случае классы - это культуры(винограда), они являются равнозначными, и нам не важно, какого рода ошибок будет больше. К тому же у нас небинарная классификация и данные равномерно распределены между классами, поэтому нам подойдет Accuracy. Она показывает количество правильно проставленных меток класса (истинно положительных и истинно отрицательных) от общего количества данных и считается следующим образом:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

8. Полученные результаты

Accuracy	skl	4 lab	Kp
Train	1.0	0.9858	0.9826
test	0.9166	0.8888	0.9655

В ходе работы удалось улучшить результаты, полученные в 4 лабораторной путем добавления регуляризации и выравнивания количество образцов по классам.