Python задание 6: регрессия и байесовские вероятности

1 Модель линейной регрессии

Пусть у нас есть набор из n кортежей вещественных чисел $\{(y_i; x_{i1}, \ldots, x_{ik})\}_{i=1}^n$, где x_{ij} считаются независимыми переменными, а y_i — зависимой переменной. Нашей целью будет научиться, используя этот набор данных, предсказывать y–и по новым x–ам.

Модель линейной регрессии предполагает, что y-и получаются как некоторая линейная комбинация x–ов:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i.$$

В этой формуле

- $\beta_0, \beta_1, \ldots, \beta_k$ — неизвестные, но постоянные коэффициенты (константы), общие для всех $\{(y_i; x_{i1}, \ldots, x_{ik})\}_{i=1}^n$.
- $\{\varepsilon\}_{i=1}^n$ представляют собой ошибки модели. Они необходимы потому, что мы не можем предполагать, что наши данные будут находиться в точной линейной зависимости. Если бы это было так, то у нас бы стояла обычная задача решения системы уравнений. Полагая, что $\{\varepsilon\}_{i=1}^n$ также являются суммой вкладов от множества других, неизвестных нам факторов, разумно будет предположить, что они имеют нормальное распределение N_{0,σ^2} с неизвестной дисперсией.

Заметим, что, так как $\beta_0, \beta_1, \ldots, \beta_k$ и $\{(x_{i1}, \ldots, x_{ik})\}_{i=1}^n$ — константы, то каждый y_i можно считать нормальной случайной величиной со средним значением $\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$ и дисперсией σ^2.

Оказывается, эти уравнения очень удобно переписать в матричной форме

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

где:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1k} \\ 1 & x_{21} & x_{22} & \ldots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nk} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

В таком случае, при выполнении ряда естественных условий, оптимальные коэффициенты $\hat{\boldsymbol{\beta}}$ можно выразить как:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},$$

где "крышечка" подчеркивает что это оценка, а не реальные $\boldsymbol{\beta}$.

После того, как мы нашли приближённые коэффициенты $\hat{\boldsymbol{\beta}}$, мы можем получить предсказания $\hat{\mathbf{y}}$ по x–ам из нашей выборки:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_k x_{ik}.$$

Эти $\hat{y}$-и содержат приближение линейной части нашей модели, зависящей от x-ов, но они не включают в себя отклонения от прочих факторов, представленные ошибками $\{\varepsilon_i\}_{i=1}^n$. Поэтому мы также можем получить приближения для этих ошибок через:

$$\hat{\varepsilon}_i = y_i - \hat{y}_i.$$

1.1 Задание № 1

В этом и во всех последующих заданиях используйте приложенный стандартный набор данных о ценах на дома в зависимости от различных факторов.

Выполните следующие шаги:

1. Исключите сильно коррелированные независимые переменные. Под «слишком высокой» корреляцией можно понимать абсолютное значение коэффициента корреляции выше 0.9. Это поможет избежать проблемы мультиколлинеарности в модели.

2. Преобразуйте категориальные переменные в числовой формат при помощи кодирования фиктивными переменными (также известного как «одноразрядное кодирование» или one-hot кодирование). Это позволит модели корректно работать с категориальной информацией.

3. Проверьте, не улучшится ли связь некоторых признаков с зависимой переменной при применении различных преобразований (например, логарифмирования или взятия квадратного корня). Если после преобразования переменной её корреляция с целевой переменной увеличивается, используйте преобразованную версию этой переменной.

1.2 Задание № 2

Напишите функцию, которая:

- Принимает на вход любой набор переменных в виде списка строк;
- Находит оценки для соответствующих параметров линейной регрессии;
- Находит приближения для ошибок линейной регрессии, а также оценку $\hat{\sigma}^2$ их дисперсии σ^2.

2 Оценка значимости переменной

Хотя для предсказания значений y нам даётся k различных переменных $\{(x_{i1}, \ldots, x_{ik})\}_{i=1}^{n}$, это не значит, что все они одинаково важны или даже вообще влияют на значения y. Мы попытаемся оставить только значимые переменные, проверяя для каждого коэффициента β_s следующие гипотезы:

$$\mathbf{H_0} : \beta_s = 0;$$

$$\mathbf{H_a} : \beta_s \neq 0.$$

Выполнение $\mathbf{H_0}$ означает, что соответствующая переменная не оказывает никакого влияния на значения y, $\mathbf{H_a}$ — наоборот, что это влияние есть.

Для этого мы посмотрим на построенные нами оценки $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$. Так как они были получены как линейные комбинации y–ов: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, то они и сами будут являться нормальными случайными величинами. Их средние значения равны реальным параметрам $\beta_0, \beta_1, \ldots, \beta_k$. А их дисперсии можно получить из матрицы ковариации $\hat{\boldsymbol{\beta}}$:

$$\begin{pmatrix} \mathbb{D}(\hat{\beta}_0) & \mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_1) & \cdots & \mathrm{Cov}(\hat{\beta}_0, \hat{\beta}_k) \\ \mathrm{Cov}(\hat{\beta}_1, \hat{\beta}_0) & \mathbb{D}(\hat{\beta}_1) & \cdots & \mathrm{Cov}(\hat{\beta}_1, \hat{\beta}_k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(\hat{\beta}_k, \hat{\beta}_0) & \mathrm{Cov}(\hat{\beta}_k, \hat{\beta}_1) & \cdots & \mathbb{D}(\hat{\beta}_k) \end{pmatrix},$$

которая, оказывается, равна $\sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$.

2.1 Задание № 3

Напишите функцию, которая, по обученной модели линейной регрессии, проводит тест на значимость любого указанного ей коэффициента $\hat{\beta}_s$. Для этого:

- Найдите оценку дисперсии $\hat{\beta}_s$, используя вместо σ^2 оценку регрессионных ошибок из предыдущего задания;
- Найдите стандартизацию коэффициента $\hat{\beta}_s^{(st)}$ при условии выполнения основной гипотезы $\mathbf{H_0}$;
- Предполагая, что стандартизация $\hat{\beta}_s^{(st)}$ имеет приблизительно стандартное нормальное распределение и что более "экстремальными" (более указывающими на альтернативную гипотезу) значениями являются большие по модулю значения, найдите p–value для этого теста.

2.2 Отбор переменных

Произведите отбор наиболее значимых переменных с помощью процедуры *backwards elimination*:

1. Сначала обучите модель используя все доступные переменные;

2. Проведите тест на значимость для каждого коэффициента и найдите соответствующее ему p–value;

3. Если все p–values оказались меньше 0.05, то алгоритм завершает работу. Иначе исключается переменная с наибольшим значением p–value, модель обучается заново с нуля и мы повторяем алгоритм с **шага 2**.

Заметим, что для категориальных переменных нельзя выкидывать отдельные категории — их значимость оценивается по их "лучшей" категории и исключаются из алгоритма все категории сразу.

3 Байесовская оценка вероятности продажи

В этом задании нас будет интересовать не сама цена, а факт продажи. В таблице данных есть бинарный признак `Sold`, который принимает значение 1, если дом был продан, и 0, если нет. Будем считать, что внутри каждой фиксированной подгруппы домов случайная величина `Sold` имеет распределение Бернулли B_p с некоторым неизвестным параметром $p \in [0, 1]$, который интерпретируем как вероятность продажи дома из этой подгруппы.

Мы будем использовать байесовский взгляд на задачу: нас интересует не просто одна оценка вероятности p, а её распределение.

3.1 Задание № 4

Напишите код на Python, который работает с исходным набором данных о домах и реализует следующую идею.

Разделите дома по типу водоёма в признаке `waterbody` (значения `None`, `River`, `Lake`, `Lake and River`).

Пользователь должен задать своё предположение о вероятности продажи дома в каждой группе (априорное математическое ожидание p), а также степень «неуверенности» в этом предположении (априорное стандартное отклонение). Считаем, что априор для p задаётся бета-распределением. Используя исходный априор и данные по каждой подгруппе, для каждой группы найдите апостериорное распределение p и выведите его математическое ожидание и стандартное отклонение. После этого сравните группы между собой: где вероятность продажи выше, где ниже, и как сильно изменилась неопределённость по сравнению с априором.

Далее нужно показать, как выглядят последовательные байесовские обновления при объединении групп. Начните с априора по предположениям пользователя о продаже домов в группе `None` и обновите его по данным этой группы, получив первый апостериор. Затем возьмите этот апостериор как новый априор и обновите его по данным второй группы, снова посчитав апостериорные среднее и стандартное отклонение. Продолжайте так в выбранном порядке, шаг за шагом объединяя группы и отслеживая, как меняются параметры распределения p после каждого обновления.

Нарисуйте графики исходного априора, заданного пользователем, и всех полученных после обновлений апостериоров, чтобы было видно, как распределение постепенно уточняется под воздействием данных.