Прикладная статистика в R

Лекция 8. Модель бинарной регрессии

Елена Михайловна Парилина

д. ф.-м. н., проф.

2021

Модуль 4. Модели нелинейной регрессии.

- Модель бинарной регрессии. Спецификация модели.
- Построение модели бинарной регрессии в R.
- Проверка значимости построенной модели.
- Оценка результатов классификации по построенной модели бинарной регрессии.
- Нарушение основных предположений регрессионного анализа.
- Ридж-регрессия.
- Медианная регрессия.
- Другие нелинейные регрессионные модели.
- Регрессионные модели в R.

Модель бинарной регрессии

Логистическая регрессия

- Логистическая регрессия (также известная как логит-регрессия или логит-модель) была разработана статистиком Дэвидом Коксом в 1958 году и представляет собой регрессионную модель, в которой зависимая переменная y является категориальной.
- Логистическая регрессия позволяет нам оценить вероятность категориального ответа на основе одной или нескольких переменных-предикторов x.
- Логистическая регрессия позволяет оценить, насколько наличие предиктора увеличивает (или уменьшает) вероятность зависимой переменной перейти в ту или иную категорию.
- В модели бинарной регрессии мы рассматриваем случай, когда y может принимать только два значения, «0» и «1», которые представляют такие результаты, как «прошел / не прошел», «выиграл / проиграл», «здоров / болен».
- Случаи, когда зависимая переменная имеет более двух категорий, могут быть проанализированы с помощью мультиномиальной логистической регрессии или, если категории упорядочены, то с помощью порядковой логистической регрессии.

Когда мы можем использовать бинарную регрессию?

В случае, когда имеется качественные (или категориальные) значения зависимой переменной, линейная регрессия не может быть применена.

Пример

Предположим, что мы пытаемся предсказать состояние здоровья пациента в отделении неотложной помощи на основе его симптомов. Пусть имеется три возможных диагноза: инсульт, передозировка наркотиками и эпилептический припадок. Мы могли бы рассмотреть возможность кодирования этих значений как количественной зависимой переменной y следующим образом:

$$y = \begin{cases} 1, & \text{если инсульт,} \\ 2, & \text{если передозировка наркотиками,} \\ 3, & \text{если эпилептический припадок.} \end{cases}$$

Почему мы будем использовать модель бинарной регрессии?

Используя это кодирование возможных диагнозов, можно попробовать использовать метод наименьших квадратов для построения модели линейной регрессии для прогнозирования y на основе набора предикторов $x_1, \ldots, x_k$.

К сожалению, это кодирование подразумевает упорядочение результатов, помещая передозировку наркотиков между инсультом и эпилептическим припадком, и настаивая на том, что разница между инсультом и передозировкой наркотиков такая же, как разница между передозировкой наркотиками и эпилептическим припадком. На практике нет никаких оснований для такого упорядочения.

Почему мы будем использовать модель бинарной регрессии?

Другое кодирование диагнозов

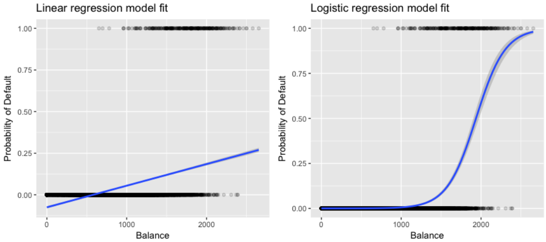
Например, можно использовать другое кодирование диагнозов:

$$y = \begin{cases} 1, & \text{если эпилептический припадок}, \\ 2, & \text{если передозировка наркотиками}, \\ 3, & \text{если инсульт}, \end{cases}$$

что означало бы совершенно другое соотношение между диагнозами. Каждая из этих кодировок создаст принципиально разные линейные модели, которые в конечном итоге приведут к различным наборам прогнозов на основе тестовых наблюдений.

Бинарная регрессия: пример

Например, если мы пытаемся классифицировать клиента как неплательщика с высоким или низким уровнем риска на основе его баланса, мы можем использовать линейную регрессию. Левый рисунок показывает, как линейная регрессия может предсказать вероятность того, что он вернет кредит.



К сожалению, для балансов, близких к нулю, мы прогнозируем отрицательную вероятность возврата кредита. Если бы мы прогнозировали для очень больших значений баланса, то мы бы получили значения возврата больше 1. Эти прогнозы неразумны, поскольку, истинная вероятность возврата кредита, независимо от баланса кредитной карты, должна находиться в диапазоне от 0 до 1.

Логистическая регрессия: пример

Чтобы избежать этой проблемы, мы должны прогнозировать y, используя функцию, которая дает выходные данные от 0 до 1 для всех значений x. Этому описанию соответствуют многие функции.

В логистической регрессии мы используем функцию, показанную на правом рисунке предыдущего слайда, которая имеет вид:

$$y(x) = \frac{e^{\beta_0+\beta_1 x_1+\ldots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\ldots+\beta_k x_k}}.$$

Бинарная регрессия (математическая модель)

$y \in \{0, 1\}$: зависимая переменная,
$x_1, \ldots, x_k$: независимые переменные (предикторы).

$x = (1, x_1, \ldots, x_k)^\mathsf{T}$

Имеется выборка: $(1, x_{i1}, \ldots, x_{ik}, y_i)$, $i = 1, \ldots, n$.

Модель бинарной регрессии:

$$\mathrm{P}\{y_i = 1\} = F(\beta^T x_i),$$

где $F(x)$ — непрерывная функция распределения.

Пусть $y_i^* = \beta^T x_i + \varepsilon_i$, где $\varepsilon_1, \ldots, \varepsilon_n$ — независимые одинаково распределенные случайные величины с математическими ожиданиями $E\varepsilon_i = 0$ и дисперсиями $D\varepsilon_i = \sigma^2$.

$$y_i = \begin{cases} 1, & y_i^* \geqslant 0, \\ 0, & y_i^* < 0. \end{cases}$$

Логит и пробит модели бинарной регрессии

Две наиболее применимые модели бинарной регрессии (в зависимости от F):

- Пробит модель: $\Phi(u) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{u} e^{-\frac{z^2}{2}} dz.$
- Логит модель: $\Lambda(u) = \frac{e^u}{1 + e^u}.$

Как найти оценки β? Рассмотрим функцию правдоподобия

$$
\begin{aligned}
L(y_1, \ldots, y_n) &= \prod_{i:y_i=0} (1 - F(\beta^T x_i)) \prod_{i:y_i=1} F(\beta^T x_i) \\
&= \prod_{i=1}^{n} F^{y_i}(\beta^T x_i)(1 - F(\beta^T x_i))^{1-y_i}, \\
\ln L(y_1, \ldots, y_n) &= \sum_{i=1}^{n} \left(y_i \ln F(\beta^T x_i) + (1 - y_i) \ln(1 - F(\beta^T x_i))\right), \\
\frac{\partial \ln L}{\partial \beta} &= \sum_{i=1}^{n} \left(\frac{y_i f(\beta^T x_i)}{F(\beta^T x_i)} - \frac{(1 - y_i) f(\beta^T x_i)}{1 - F(\beta^T x_i)} \right) x_i = 0.
\end{aligned}
$$

Функция f — плотность распределения, построенная по функции F.

Для логит модели:

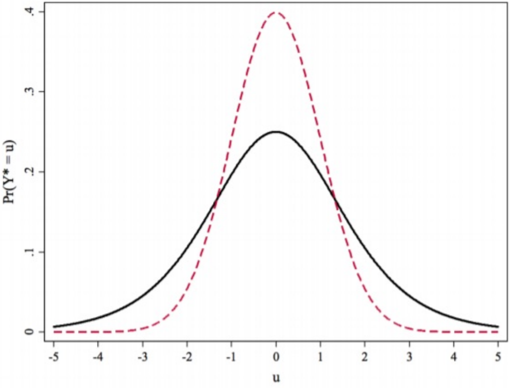
$$\frac{\partial \ln L}{\partial \beta} = \sum_{i=1}^{n} (y_i - \Lambda(\beta^T x_i)) x_i = 0.$$

Для пробит модели:

$$\frac{\partial \ln L}{\partial \beta} = \sum_{i=1}^{n} \frac{(2y_i - 1)\varphi(\beta^T x_i)}{\Phi((2y_i - 1)\beta^T x_i)} x_i = 0, \quad \varphi(x) = \Phi'(x).$$

Логит vs пробит

Графики логистической плотности распределения (черн.) и плотности нормального распределения (красн.):



Построение логит модели в R

Логистическая регрессия: данные файла "mydata.Rdata"

	admit	gre	gpa	rank
1	0	380	3.61	3
2	1	660	3.67	3
3	1	800	4.00	1
4	1	640	3.19	4
5	0	520	2.93	4
6	1	760	3.00	2
7	1	560	2.98	1
8	0	400	3.08	2
9	1	540	3.39	3
10	0	700	3.92	2

- Имеется 4 переменные: принят или нет в университет (`admit`), оценка на экзамене (`gre`), средний балл в школе (`gpa`), ранг школы (`rank`).
- Переменные `gre` и `gpa` являются непрерывными, переменная `rank` дискретна (значения от 1 (лучшая школа) до 4 (худшая школа) в рейтинге).

Логистическая регрессия: описательная статистика

- При необходимости составьте описательную статистику, постройте графики.

Функции в R

```
summary(mydata)
```

или

```
sapply(mydata, sd)
```

```
> summary(mydata)
     admit             gre             gpa             rank      
 Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000  
 1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000  
 Median :0.0000   Median :580.0   Median :3.395   Median :2.000  
 Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485  
 3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000  
 Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000  
> sapply(mydata, sd)
      admit         gre         gpa        rank 
  0.4660867 115.5165364   0.3805668   0.9444602 
```

- Для построения таблицы сопряженности категориальной переменной `admit` и предикторов типа `rank` мы можем использовать следующие функции:

Функции в R

```
xtabs(~ admit + rank, data = mydata)
```

```
> xtabs(~admit + rank, data = mydata)
     rank
admit  1  2  3  4
    0 28 97 93 55
    1 33 54 28 12
```

Логистическая регрессия: базовая модель

- Построим базовую модель логистической регрессии, используя максимально возможное число (разумных) переменных. Синтаксис функции `glm` аналогичен синтаксису функции `lm`, за исключением того, что мы должны передать аргумент `family = binomial`, чтобы указать R использовать логистическую регрессию, а не какой-либо другой тип регрессионной модели.

Функции в R

```
mydata$rank <- factor(mydata$rank)
mylogit <- glm(admit ~ gre + gpa + rank, data = mydata, family =
binomial("logit"))
summary(mylogit)
```

- Мы используем функцию `factor`, чтобы сделать эту переменную категориальной.
- `admit ~ gre + gpa + rank` — формула регрессии.
- Значение "binomial" может быть "logit" , "probit" , "cauchit".
- Функциия `glm` использует метод максимального правдоподобия для нахождения оценок.

Логистическая регрессия: базовая модель

```
Call:
glm(formula = admit ~ gre + gpa + rank, family = "binomial",
    data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6268  -0.8662  -0.6388   1.1490   2.0790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.989979   1.139951  -3.500 0.000465 ***
gre          0.002264   0.001094   2.070 0.038465 *
gpa          0.804038   0.331819   2.423 0.015388 *
rank2       -0.675443   0.316490  -2.134 0.032829 *
rank3       -1.340204   0.345306  -3.881 0.000104 ***
rank4       -1.551464   0.417832  -3.713 0.000205 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.52  on 394  degrees of freedom
AIC: 470.52

Number of Fisher Scoring iterations: 4
```

Анализ базовой модели

Анализ базовой модели

- Проведем анализ базовой модели:
  - Построенная модель логистической регрессии:

$$P(admit = 1) = \frac{e^{\beta x}}{1 + e^{\beta x}},$$

  где $\beta x = -3.989979 + 0.002264 \cdot gre + 0.804038 \cdot gra - 0.675443 I\{rank = 2\} - 1.340204 I\{rank = 3\} - 1.551464 I\{rank = 4\}$.
  - Коэффициенты логистической регрессии показывают изменение логарифма вероятности того, что зависимая переменная примет значение 1 при увеличении на единицу переменной-предиктора.
  - На каждую единицу изменения в "gre" логарифмическая вероятность поступления (по сравнению с непоступлением) увеличиваются на 0.002.
  - При увеличении "gpa" на одну единицу логарифмическая вероятность поступления увеличиваются на 0.804.
  - Индикаторные переменные для ранга имеют несколько иное толкование. Например, посещение школы с рейтингом 2 по сравнению со школой с рейтингом 1 изменяет логарифмические шансы поступления на -0.675.

Анализ базовой модели

- Мы анализируем статистику Вальда (нулевая гипотеза H_0: $\beta_i = 0$ при альтернативой гипотезе H_1: $\beta_i \neq 0$), которая в таблице называется z-значением. Если соответствующие p-value меньше 0.05, мы отклоняем нулевую гипотезу и утверждаем, что коэффициент значим.
- В нашем примере все коэффициенты значимы для уровня 0.05.
- AIC $= 470.52$ — это значение информационного критерия Акаике. Мы можем использовать его для сравнения моделей. Чем меньше значение, тем лучше модель.
- Критерий максимального правдоподобия — это проверка значимости модели в целом (H_0: все $\beta_i = 0$). Этот тест сравнивает две модели (модель с предикторами и модель только со свободным членом). Статистика критерия — это разница между отклонением остатков для модели с предикторами и нулевой моделью (499.98-458.52). Статистика критерия подчиняется распределению хи-квадрат с количеством степеней свободы, равным разнице в степенях свободы между текущей и нулевой моделями (то есть, количество переменных-предикторов в модели), то есть 399-394.

Анализ базовой модели

- Значение p-value для этой статистики можно посчитать следующим образом:

Функции в R

```
with(mylogit, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
```

```
> with(mylogit, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 7.578194e-08
```

p-value меньше 0.05, следовательно, базовая модель логистической регрессии значима в целом.

Анализ базовой модели

- Критерий Вальда предназначен для проверки значимости отдельных коэффициентов регрессии или модели в целом (H_0: все $\beta_i = 0$ или некоторые $\beta_i = 0$). В `Terms` приводится список коэффициентов при независимых переменных для тестирования.

Функции в R

```
install.packages("aod")
library(aod)
wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 2:6)
```

В помощью аргумента `Terms` мы выдаем список переменных для тестирования.

```
> wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 2:6)
Wald test:
----------

Chi-squared test:
X2 = 36.1, df = 5, P(> X2) = 8.9e-07
```

Значение p-value меньше 0.05, тогда коэффициенты $\beta_1, \ldots, \beta_5$ значимы, или модель значима в целом.

Анализ базовой модели

- Чтобы построить доверительные интервалы, мы можем использовать функцию `confint`. Обратите внимание, что для логистических моделей доверительные интервалы строятся на основании функции логарифма правдоподобия. Мы также можем получить доверительные интервалы, основанные только на стандартных ошибках (функция `confint.default`).

Функции в R

```
confint(mylogit)
confint.default(mylogit)
```

Анализ базовой модели

Доверительные интервалы:

```
> confint(mylogit)
Waiting for profiling to be done...
                    2.5 %       97.5 %
(Intercept) -6.2716202334 -1.792547080
gre          0.0001375921  0.004435874
gpa          0.1602959439  1.464142727
rank2       -1.3008888002 -0.056745722
rank3       -2.0276713127 -0.670372346
rank4       -2.4000265384 -0.753542605
> confint.default(mylogit)
                    2.5 %       97.5 %
(Intercept) -6.2242418514 -1.755716295
gre          0.0001202298  0.004408622
gpa          0.1536836760  1.454391423
rank2       -1.2957512650 -0.055134591
rank3       -2.0169920597 -0.663415773
rank4       -2.3703986294 -0.732528724
```

Пробит модель

Пробит vs логит модель

Используя нормальное распределение в модели бинарной регрессии, мы строим **пробит модель**.

Функции в R

```
myprobit <- glm(admit ~ gre + gpa + rank, data = mydata, family
= binomial("probit"))
summary(myprobit)
```

- AIC=470.41, пробит модель немного лучше в смысле коэф. AIC.
- Все коэффициенты значимы при уровне 0.05.
- Для пробит модели статистика максимального правдоподобия равна 41.57, значение *p*-value меньше 0.05, т.е. пробит модель в целом значима.
- Критерий Вальда: *p*-value также меньше 0.05, т.е. в целом пробит модель значима.

```
> with(myprobit, pchisq(null.deviance - deviance, df.null - df.residual, lower.tail = FALSE))
[1] 7.218932e-08
> wald.test(b = coef(myprobit), Sigma = vcov(mylogit), Terms = 2:6)
Wald test:
----------

Chi-squared test:
X2 = 13.1, df = 5, P(> X2) = 0.022
```

Анализ пробит модели

```
> myprobit <- glm(admit ~ gre + gpa + rank, data = mydata, family = binomial("probit"))
> summary(myprobit)

Call:
glm(formula = admit ~ gre + gpa + rank, family = binomial("probit"),
    data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6163  -0.8710  -0.6389   1.1560   2.1035

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.386836   0.673946  -3.542 0.000398 ***
gre          0.001376   0.000650   2.116 0.034329 *
gpa          0.477730   0.197197   2.423 0.015410 *
rank2       -0.415399   0.194977  -2.131 0.033130 *
rank3       -0.812138   0.208358  -3.898 9.71e-05 ***
rank4       -0.935899   0.245272  -3.816 0.000136 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.41  on 394  degrees of freedom
AIC: 470.41

Number of Fisher Scoring iterations: 4
```

Диагностика модели бинарной регрессии

Диагностика модели бинарной регрессии. Способность правильно классифицировать данные

При разработке моделей для прогнозирования наиболее важной метрикой является оценка того, насколько хорошо модель предсказывает целевую переменную при наблюдениях за пределами выборки. Во-первых, нам нужно использовать построенные модели для прогнозирования значений на первоначальном наборе обучающих данных (обучение модели).

Функции в R

```
test.predicted.logit <- predict(mylogit, newdata = mydata, type = "response")
test.predicted.probit <- predict(myprobit, newdata = mydata, type = "response")
```

Таблица сопряженности

Предсказания vs Наблюдения	1	0
1	TP	FP
0	FN	TN

- TP: True Positive,
- FP: False Positive,
- FN: False Negative,
- TN: True Negative.

Следующая функция классифицирует данные, используемые для построения модели логистической регрессии, прогнозируя зависимую переменную, когда вероятность для прогноза зависимой переменной равна 0.5.

Функции в R

```
install.packages("InformationValue")
library(InformationValue)
confusionMatrix(mydata$admit, test.predicted.logit, threshold =
0.5)
```

Таблицы сопряженности

```
> confusionMatrix(mydata$admit, test.predicted.logit,threshold = 0.5)
    0  1
0 254 97
1  19 30
> confusionMatrix(mydata$admit, test.predicted.probit,threshold = 0.5)
    0  1
0 254 97
1  19 30
> confusionMatrix(mydata$admit, test.predicted.logit,threshold = 0.7)
    0   1
0 272 125
1   1   2
> confusionMatrix(mydata$admit, test.predicted.probit,threshold = 0.7)
    0   1
0 272 125
1   1   2
```

Специфичность и чувствительность

Предсказания vs Наблюдения	1	0
1	TP	FP
0	FN	TN

Чувствительность модели (или True Positive Rate) — это доля правильно предсказанных единиц среди всех предсказанных единиц:

$$Sensitivity = \frac{TP}{TP + FN}.$$

Специфичность модели — это доля правильно предсказанных нулей среди всех наблюдаемых нулей:

$$Specificity = \frac{TN}{TN + FP}.$$

Специфичность и чувствительность

Функции в R

```
sensitivity(mydata$admit, test.predicted.logit, threshold = 0.5)
specificity(mydata$admit, test.predicted.probit, threshold = 0.5)
```

```
> sensitivity(mydata$admit, test.predicted.logit, threshold = 0.5)
[1] 0.2362205
> sensitivity(mydata$admit, test.predicted.probit, threshold = 0.5)
[1] 0.2362205
> specificity(mydata$admit, test.predicted.probit, threshold = 0.5)
[1] 0.9304029
> specificity(mydata$admit, test.predicted.logit, threshold = 0.5)
[1] 0.9304029
```

Выбор «оптимального» порогового значения вероятности для предсказания

- По умолчанию порог вероятности предсказания составляет 0.5. Но иногда настройка порога вероятности может повысить точность как при разработке модели, так и при ее валидации.
- Функция `InformationValue :: optimCutoff` предоставляет способы нахождения оптимального порога, чтобы улучшить предсказание единиц, нулей, одновременно единиц и нулей, а также уменьшить ошибку неправильной классификации.
- Мы можем вычислить значения критерия, который минимизирует ошибку неправильной классификации для построенной модели.

Функции в R

```
library(InformationValue)
optCutOff <- optimalCutoff(mydata$admit,
test.predicted.logit)[1]
confusionMatrix(mydata$admit, test.predicted.logit, threshold =
optCutOff)
```

«Оптимальное» пороговое значение вероятности

```
> optCutOff <- optimalCutoff(mydata$admit, test.predicted.logit)[1]
> optCutOff
[1] 0.4684082
> confusionMatrix(mydata$admit, test.predicted.logit,threshold = optCutOff)
    0  1
0 247 89
1  26 38
> sensitivity(mydata$admit, test.predicted.logit, threshold = optCutOff)
[1] 0.2992126
```

Разделение выборки на тестовую и тренировочную

Мы можем разделить выборку на тренировочную и тестовую с помощью функции `sample_frac( )` из библиотеки "dplyr".
Следующая процедура разделит выборку на две части в соотношении 70% — 30% на тренировочную и тестовую:

Функции в R

```
install.packages("dplyr")
library(dplyr)
train < − sample_frac(mydata, 0.7)
sample_id <- as.numeric(rownames(train)) # rownames() возвращает
номера элементов выборки типа as.numeric
test <- mydata[-sample_id,]
```

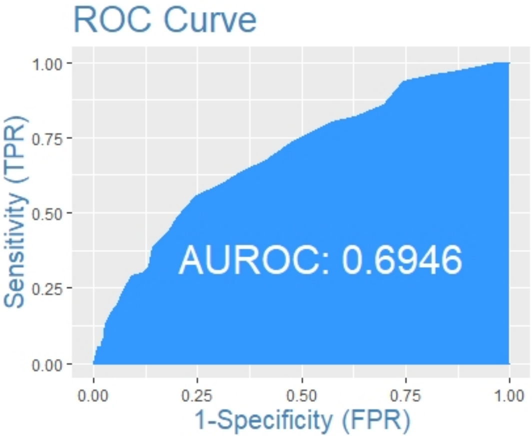
ROC кривая

- Кривая ROC (или Receiver Operating Characteristic) — еще один распространенный инструмент, используемый с бинарными классификаторами.
- Характеристика ROC суммирует производительность модели, оценивая компромисс между частотой истинных положительных результатов (чувствительность) и частотой ложных положительных результатов (1-специфичность).
- Для построения графика ROC кривой рекомендуется принять $p > 0.5$, поскольку мы стремимся в модели увеличить вероятность успешных предсказаний. ROC суммирует предсказательную силу для всех возможных значений $p > 0.5$.
- Площадь под кривой (AUC), называемая индексом точности (accuracy) или индексом соответствия (concordance index), является хорошей метрикой для кривой ROC. Чем больше площадь под кривой, тем выше точность прогноза модели.
- AUC идеальной прогнозной модели равна 1.

ROC кривая

Функции в R

```
plotROC(mydata$admit, test.predicted.logit)
```



Выбор наилучшей модели бинарной регрессии

Выбор наилучшей модели с помощью AIC

Мы можем использовать **информационный критерий Акаике (AIC)** в качестве критерия «лучшей» модели логистической или пробит модели регрессии. Модель с **более низким показателем AIC** лучше.

Значение AIC модели:

$$AIC = 2k - 2\ln \hat{L},$$

где k — число оцениваемых параметров модели, $\hat{L}$ — максимальное значение функции правдоподобия модели.

Функции в R

```
library(MASS)
stepAIC(mylogit)
stepAIC(myprobit)
```

`stepAIC(mylogit, direction=c("both" ,"backward" ,"forward"))` возвращает наилучшую модель.

Выбор наилучшей логит модели

```
> stepAIC(mylogit)
Start:  AIC=470.52
admit ~ gre + gpa + rank

       Df Deviance    AIC
<none>      458.52 470.52
- gre   1   462.88 472.88
- gpa   1   464.53 474.53
- rank  3   480.34 486.34

Call:  glm(formula = admit ~ gre + gpa + rank, family = "binomial",
    data = mydata)

Coefficients:
(Intercept)          gre          gpa        rank2        rank3        rank4
  -3.989979     0.002264     0.804038    -0.675443    -1.340204    -1.551464

Degrees of Freedom: 399 Total (i.e. Null);  394 Residual
Null Deviance:      500
Residual Deviance: 458.5        AIC: 470.5
```

Выбор наилучшей пробит модели

```
> stepAIC(myprobit)
Start:  AIC=470.41
admit ~ gre + gpa + rank

       Df Deviance    AIC
<none>      458.41 470.41
- gre   1   462.96 472.96
- gpa   1   464.44 474.44
- rank  3   480.19 486.19

Call:  glm(formula = admit ~ gre + gpa + rank, family = binomial("probit"),
    data = mydata)

Coefficients:
(Intercept)          gre          gpa        rank2        rank3        rank4
  -2.386836     0.001376     0.477730    -0.415399    -0.812138    -0.935899

Degrees of Freedom: 399 Total (i.e. Null);  394 Residual
Null Deviance:      500
Residual Deviance: 458.4        AIC: 470.4
```

План отчета о построенной модели бинарной регрессии

1. Построить описательную статистику с помощью функции `summary`.
2. При необходимости создать категориальную переменную (функция `factor`).
3. Построить базовую модель логистической регрессии с максимально возможным количеством предикторов с использованием функции `glm`.
4. Записать уравнение бинарной регрессии, используя оценки коэффициентов.
5. Протестировать значимость коэффициентов регрессии в отдельности с использованием функции `summary(model)`.
6. Проверить значимость регрессии в целом по критерию Вальда и максимального правдоподобия (функции `summary(model)` и `wald.test`).
7. Построить доверительные интервалы для коэффициентов регрессии (функции `confint` и `confint.default`).
8. Провести сравнительный анализ логит и пробит моделей.

9. Построить таблицу сопряженности с пороговой вероятностью 0.5 (функция `confusionMatrix`).
10. Посчитать специфичность и чувствительность модели (функции `sensitivity` и `specificity`).
11. Найти оптимальное пороговое значение вероятности предсказания. Построить таблицу сопряженности для этой вероятности, посчитать специфичность и чувствительность модели (функция `optimalCutoff`).
12. Если выборка большого объема, то можно предварительно разбить выборку на две части: тренировочная и тестовая выборки (функция `sample_frac`).
13. Построить ROC кривую, интерпретировать результаты (функция `plotROC`).
14. Попробовать улучшить логит или пробит модель с использованием коэффициента AIC (функция `stepAIC`).

Итоги

Что мы узнали на Лекции 8?

- Мы узнали, что такое модель бинарной регрессии.
- Мы узнали, как построить модель бинарной регрессии в R.
- Мы узнали, как проверять значимость построенной модели.
- Мы узнали, как оценить результаты классификации по построенной модели бинарной регрессии.

Что мы узнаем на Лекции 9?

Мы узнаем,

- какие еще регрессионные модели существуют.
- как построить медианную регрессию.
- как построить ридж-регрессию и зачем это нужно.
- какие нелинейные модели регрессии существуют.

Спасибо за внимание и до встречи на Лекции 9!